OIPE JC134 JUN 11 2003 PATENT & TRADEMARK OFFICE

06-12-03

AF

9700

PTO/SB/21 (01-03)
Approved for use through 04/30/2003. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09 / 346,354
Filing Date	02-Jul-1999
First Named Inventor	Saeed Ganji
Art Unit	2624
Examiner Name	King Y. Poon
Attorney Docket Number	EFIM051

RECEIVED
JUN 17 2003
Technology Center 2600

Total Number of Pages in This Submission

ENCLOSURES (*Check all that apply*)

- [x] Fee Transmittal Form
- [x] Fee Attached
- [] Amendment/Reply
 - [] After Final
 - [] Affidavits/declaration(s)
- [] Extension of Time Request
- [] Express Abandonment Request
- [] Information Disclosure Statement
- [] Certified Copy of Priority Document(s)
- [] Response to Missing Parts/ Incomplete Application
 - [] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [] Drawing(s)
- [] Licensing-related Papers
- [] Petition
- [] Petition to Convert to a Provisional Application
- [] Power of Attorney, Revocation Change of Correspondence Address
- [] Terminal Disclaimer
- [] Request for Refund
- [] CD, Number of CD(s) _____
- [] After Allowance Communication to Group
- [] Appeal Communication to Board of Appeals and Interferences
- [x] Appeal Communication to Group **(Appeal Notice, Brief, Reply Brief)**
- [] Proprietary Information
- [] Status Letter
- [x] Other Enclosure(s) (please Identify below):
 1. Appeal Brief (3 copies)
 2. Fee Transmittal (2 copies)
 3. Return Postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual	James Trosino
Signature	James Trosino
Date	11-Jun-2003

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231 on this date: 11-Jun-2003

Typed or printed	James Trosino		
Signature	James Trosino	Date	11-Jun-2003

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, Washington, DC 20231.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

OIPE JC194 JUN 11 2003 PATENT & TRADEMARK OFFICE

PTO/SB/17 (01-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2003

Effective 01/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 320.00

Complete if Known	
Application Number	09 / 346,354
Filing Date	02-Jul-1999
First Named Inventor	Saeed Ganji
Examiner Name	King Y. Poon
Art Unit	2624
Attorney Docket No.	EFIM051

RECEIVED JUN 17 2003 Technology Center 2600

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☑ Deposit Account:

Deposit Account Number: 05-0770

Deposit Account Name: Electronics for Imaging, Inc.

The Commissioner is authorized to: *(check all that apply)*

☑ Charge fee(s) indicated below ☑ Credit any overpayments

☑ Charge any additional fee(s) during the pendency of this application

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	750	2001	375	Utility filing fee	
1002	330	2002	165	Design filing fee	
1003	520	2003	260	Plant filing fee	
1004	750	2004	375	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

	Extra Claims		Fee from below	Fee Paid
Total Claims		-20** =	X $18.00	=
Independent Claims		- 3** =	X $84.00	=
Multiple Dependent				=

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	84	2201	42	Independent claims in excess of 3
1203	280	2203	140	Multiple dependent claim, if not paid
1204	84	2204	42	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	410	2252	205	Extension for reply within second month	
1253	930	2253	465	Extension for reply within third month	
1254	1,450	2254	725	Extension for reply within fourth month	
1255	1,970	2255	985	Extension for reply within fifth month	
1401	320	2401	160	Notice of Appeal	
1402	320	2402	160	Filing a brief in support of an appeal	320.00
1403	280	2403	140	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,300	2453	650	Petition to revive - unintentional	
1501	1,300	2501	650	Utility issue fee (or reissue)	
1502	470	2502	235	Design issue fee	
1503	630	2503	315	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	750	2809	375	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	750	2810	375	For each additional invention to be examined (37 CFR 1.129(b))	
1801	750	2801	375	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 320.00

SUBMITTED BY				(Complete *if applicable*)	
Name (Print/Type)	James Trosino	*Registration No. (Attorney/Agent)*	39,862	*Telephone*	(650) 357-3997
Signature	James Trosino			*Date*	11-Jun-2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, Washington, DC 20231.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

OIPE JC134 JUN 17 2003 PATENT & TRADEMARK OFFICE

1003

#24
6-27-03

PATENTS
EFIM0051

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

RECEIVED
JUN 17 2003
Technology Center 2600

Appellant	:	Saeed Ganji
Application No.	:	09/346,354
Filed	:	2 July 1999
For	:	PRINTER DESCRIPTION FILE FORMAT GENERATOR
Group Art Unit	:	2624
Examiner	:	KING Y. POON

APPEAL BRIEF

Appellant submits this brief pursuant to the appeal from the Examiner's final rejection dated 25 November 2002.

06/17/2003 [illegible] 00000017 09346354
01 FC:1402 320.00 OP

Table of Contents

Table of Authorities	ii
Real Party In Interest	1
Related Appeals And Interferences	2
Status Of Claims	3
Status Of Amendments	4
Summary Of Invention	5
Issue	7
Grouping Of Claims	8
Argument	9
Summary of Independent Claims	9
Summary of Examiner's Rejections	9
Discussion of deSilva	10
Discussion of Andrews	11
Discussion of the § 103 Rejections	11
Conclusion	14
APPENDIX A	A-1
APPENDIX B	B-1
APPENDIX C	C-1

Table of Authorities

Ecolochem, Inc. v. Southern Calif. Edison Co., 227 F.3d 1361 (Fed. Cir. 2000)

Crown Oper. Int'l Ltd. v. Solutia Inc., 289 F.3d 1367 (Fed. Cir. 2002)

In re Thrift, 298 F.3d 1357 (Fed. Cir. 2002)

In re Sang-Su Lee, 277 F.3d 1338 (Fed. Cir. 2002)

Real Party In Interest

The real party in interest is assignee, Electronics for Imaging, Inc., 303 Velocity Way, Foster City, CA 94404.

Related Appeals And Interferences

None.

Status Of Claims

Claims 1-4, 9 and 11	Rejected
Claims 5-8, 10	Canceled

Status Of Amendments

Amendment Mailed 27 January 2003 Not Entered

Summary Of Invention

This invention pertains to PostScript® Printer Description ("PPD") files, and more particularly to methods and apparatus for generating PPD files. A PPD file includes ASCII text that describes features of a printer, and also includes text strings used in user-interface dialogs used by a PostScript printer driver. PPD files may be used in Windows, MAC-OS, and UNIX software environments, and may be used by many software applications. One problem frequently encountered when creating PPD files, however, is that a PPD file may work correctly for one printer driver, but may fail when used by another printer driver. For example, the text strings in the PPD file must be translated for all supported languages (e.g., English, French, German, Japanese). However, a printer driver on one software platform may interpret non-English characters differently than a printer driver on a second software platform. As a result, when creating a PPD file for a particular printer, a developer typically must provide multiple PPD files for the printer, each one of which is suited for a particular software environment and a particular language. Previously known techniques for generating multiple PPD files for different software platforms and languages involve a significant amount of manual typing and copying, both of which are time-consuming and prone to error.

In accordance with this invention, apparatus and methods are provided for producing PPD text files. In particular, a development environment is provided for producing a PPD text file. (Page 5, lines 15-18; FIG. 2). The development environment includes a base PPD text file that has information regarding a printer, including text in a first language. (Page 5, lines 18-28; Page 5, lines 34-38; Page 6, lines 1-9; Page 9, lines 32-33; FIG. 2). A PPD generator is adapted to import the base PPD text file and to generate a second PPD text file that includes text translated from the first language to a second language. (Page 5, lines 28-32; Page 6, lines 9-10; Page 6, line 25 through Page 7, line 8; Page 9, lines 10-25; FIG. 2).

In addition, methods are provided for generating or revising a PPD text file. The methods include providing a base PPD text file that is adapted for a first software platform and that includes information regarding a printer, including text in a first language. (Page 5, lines 18-28; Page 5, lines 34-38; Page 6, lines 1-9; Page 9, lines 32-33; FIG. 2). The methods also include providing a build file that has information about

how the base PPD text file should be edited to provide a second PPD text file that includes text translated from the first language to a second language. (Page 6, lines 12-23; Page 9, lines 9-10; Page 9, lines 35-37). The methods further include implementing the build file to generate the second PPD text file. (Page 7, lines 10-18; Page 7, lines 25-36; Page 9, lines 2-5; FIG. 2).

Issue

Whether claims 1-4, 9 and 11 are unpatentable under 35 U.S.C. § 103(a) as obvious over deSilva U.S. Patent No. 5,857,064 ("deSilva") in view of Andrews et al. U.S. Patent No. 5,768,564 ("Andrews").

Grouping Of Claims

For purposes of this appeal, the group of claims 1-4, 9 and 11 stand or fall together.

Argument

The Examiner has rejected claims 1-4, 9 and 11 under 35 U.S.C. § 103(a) as obvious over deSilva in view of Andrews. Appellant disagrees, and respectfully requests that the Board reverse the decision of the Examiner.

Summary of Independent Claims

Amended independent claim 1 recites a development environment for producing a PPD text file, the development environment comprising a base PPD text file that comprises information regarding a printer, the information including text in a first language, and a PPD generator adapted to import the base PPD text file and to generate therefrom a second PPD text file that includes text translated from the first language to a second language. Amended independent claims 3 and 4 recite methods for generating or revising a PPD text file, the methods comprising providing a base PPD text file that comprises information regarding a printer, the information including text in a first language, the base PPD text file adapted for a first software platform, providing a build file that comprises information as to how the base PPD text file should be edited to provide a second PPD text file that includes text translated from the first language to a second language, and implementing the build file to generate the second PPD text file.

Summary of Examiner's Rejections

With regard to independent claim 1, the Examiner asserts that deSilva teaches many of the claim elements, but does not describe a developer that uses a PPD generator to import a base PPD and generate therefrom a second PPD text file that includes text translated from a first language to a second language. The Examiner asserts, however, that Andrew "in the same area of developing computer programs for computers to execute" teaches that a programmer may write and maintain a computer program in a source language and use a translator to translate the source code to other languages. The Examiner further asserts that it would have been obvious to modify deSilva to use translators to import a base PPD and generate therefrom a second PPD text file that includes text translated from a first language to a second language because

"it would save time and effort for the PPD file developer by avoiding rewriting complex and difficult programs in multiple languages to run on multiple brands of computers."

With regard to independent claims 3 and 4, the Examiner asserts that deSilva teaches many of the claim elements, but does not describe translating text from a first language to a second language. The Examiner asserts, however, that Andrew "in the same area of developing computer programs for computers to execute" teaches that rewriting computer programs to run in multiple languages is impractical, and that a programmer may write and maintain a computer program in a source language and use a translator to translate the source code to other languages. The Examiner further asserts that it would have been obvious to modify deSilva to translate text from a first language to a second language, once again because "it would save time and effort for the PPD file developer by avoiding rewriting complex and difficult programs in multiple languages to run on multiple brands of computers."

Discussion of deSilva

deSilva describes an imaging system that provides a facility for converting complex graphical primitives that cannot be directly imaged by a page description language ("PDL"), such as PostScript, into a plurality of less complex graphical primitives that can be directly imaged by the PDL. (Abstract). The system includes a printer handler, which is "a type of printer driver that controls and drives a specific printer; its purpose is to convert textual and graphical objects into printer readable form for any particular printer type." (Col. 10, lines 42-45). The handler "reads" the printer's "personality document" to access personality objects, such as the printer configuration and resolution, print job description, and the print device object that converts graphic data into the printer's imaging model. (Col. 12, lines 36-51). A personality document "is created for each type of printer that a printer handler is created for." (Col. 13, lines 28-29). By way of example, deSilva lists selected entries from a PPD file to illustrate features of the PPD that can be used to determine attributes of a PostScript printer. (Col. 18, line 29 through Col. 19, line 5).

Although deSilva includes a brief discussion of the use of information in a PPD file, deSilva does not describe or suggest anything else related to the claimed invention. First, deSilva does not describe or suggest a development environment for

producing a PPD text file, or methods for generating or revising a PPD text file. deSilva nowhere describes or suggest methods or apparatus for producing such files. Additionally, deSilva does not describe or suggest providing a development environment that includes a base PPD text file that comprises information regarding a printer, the information including text in a first language, the base PPD text file adapted for a first software platform.

Moreover, deSilva does not describe or suggest a PPD generator adapted to import the base PPD text file and to generate therefrom a second PPD text file that includes text translated from the first language to a second language. Further, deSilva does not describe or suggest methods for generating or revising a PPD text file, the methods comprising providing a build file that comprises information as to how the base PPD text file should be edited to provide a second PPD text file that includes text translated from the first language to a second language. Indeed, deSilva contains no discussion or suggestion related to any type of file that includes information as to how a base PPD text file should be edited to provide a second PPD text file that includes text translated from a first language to a second language.

Discussion of Andrews

Andrews describes methods and apparatus for automatically translating software source code from one high-level programming language to source code in another high-level programming language. (Col. 2, lines 56-59; Col. 4, lines 13-18). In particular, Andrews describes the "Rosetta Translator," that translates software source code from "portable Transaction Application Language (pTAL) to C++." (Col. 4, lines 18-22). Andrews does not, however, describe anything related to PPD text files, or a PPD generator adapted to import a base PPD text file and to generate therefrom a second PPD text file that includes text translated from the first language to a second language. Indeed, Andrews does not describe or suggest anything relevant to the claimed invention.

Discussion of the § 103 Rejections

A claimed invention is unpatentable as obvious "if the differences between the subject matter sought to be patented and the prior art are such that the subject matter

as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." §103(a). An obviousness determination may not be based on hindsight. See Ecolochem, Inc. v. Southern California Edison Co., 227 F.3d 1361, 1371 (Fed. Cir. 2000) (hindsight reconstruction may not be used "to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.") (citation omitted).

Ignoring this prohibition, the Examiner has selectively excised individual words and phrases from deSilva to reconstruct elements of the claimed invention. Indeed, the Examiner first selectively excised the words "developer," "PostScript," and "PPD," that appear randomly throughout deSilva as evidence that deSilva describes a development environment for producing a PPD text file associated with a printer. A claim is not rendered obvious, however, simply because the claim language includes some of the same words and phrases scattered throughout the text of a reference. Crown Oper. Int'l Ltd. v. Solutia Inc., 289 F.3d 1367, 1376 (Fed. Cir. 2002) ("Determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention.").

Moreover, deSilva uses the selected words and phrases in a completely unrelated context to that of the claimed invention. The Examiner cites col.12, line 48 and col. 13, lines 29-32 for the word "developer." Read in its proper context, both citations pertain to a printer manufacturer that "develops" (i.e., manufactures) a printer. (Col. 12, lines 37-49; col. 13, lines 28-32). Neither reference to the word "developer" pertains in any way to the claimed invention, i.e., a development environment for producing a PPD text file. Next, the Examiner cites col. 18, line 32 for the word "PPD." Read in its proper context, this reference states that features of a PPD "can be used to determine a PostScript printer's attributes." (Col. 18, lines 30-33). Once again, this reference does not pertain in any way to a development environment for producing a PPD text file.

Nevertheless, after conceding that deSilva does not describe or suggest the claimed invention, the Examiner cites Andrew "in the same area of developing computer programs for computers to execute." Contrary to the Examiner's assertion, there is no support for such the assertion that deSilva is related to "developing computer programs for computers to execute." In contrast, as described above, deSilva pertains to

an imaging system that provides a facility for converting complex graphical primitives that cannot be directly imaged by a PDL into a plurality of less complex graphical primitives that can be directly imaged by the PDL.

Further, there is no objective evidence that a person of ordinary skill in the relevant art would be motivated to combine deSilva and Andrews. Where obviousness is based on a combination of references, there must be an objective teaching, motivation, or suggestion to select and combine the references. See In re Thrift, 298 F.3d 1357, 1363 (Fed. Cir. 2002) (A prima facie case of obviousness requires "some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art [that] would lead an individual to combine the relevant teachings of the references."); Crown, 289 F.3d 1367, 1376 (Fed. Cir. 2002) ("[T]here must be a teaching or suggestion within the prior art, within the nature of the problem to be solved, or within the general knowledge of a person of ordinary skill in the field of the invention to look to particular sources, to select particular elements, and to combine them as combined by the inventor."); In re Sang-Su Lee, 277 F.3d 1338, 1343 (Fed. Cir. 2002) ("[T]he search for and analysis of the prior art includes evidence relevant to the finding of whether there is a teaching, motivation, or suggestion to select and combine the references relied on as evidence of obviousness.").

Instead, using hindsight gleaned from appellant's specification, the Examiner has impermissibly attempted to cobble together the claimed invention by combining the disparate teachings of deSilva and Andrews. Indeed, the Examiner has failed to provide any objective evidence of a teaching, motivation, or suggestion to select and combine deSilva and Andrews. Instead of identifying such objective evidence, the Examiner summarily concludes, with no support, that it would have been obvious "to have modified the PPD files developing method of deSilva by the teaching of Andrew because . . . it would save time and effort for the PPD file developer by avoiding rewriting complex and difficult programs in multiple languages to run on multiple brands of computers." Such a conclusory statement, unsupported by any objective evidence, is impermissible, and fails to establish any teaching, suggestion, or motivation to combine the references. See Sang-Su Lee, 277 F.3d at 1343-44 (examiner's conclusory statements suggesting reasons for combining references fail to

adequately address the issue of motivation to combine, which cannot be based on subjective belief and unknown authority).

Appellant respectfully submits that there is no such objective evidence, and that there also is no conceivable reason why a person of ordinary skill in the art would somehow be motivated to combine these unrelated references. deSilva is concerned with an object-oriented printing interface, and Andrews relates to high-level software source code language conversion. There is no objective evidence indicating that a person of ordinary skill in the art would somehow be motivated to combine deSilva and Andrews. Further, even if there were some suggestion or motivation to combine deSilva with Andrews, it is unclear how the two references could be combined to produce anything that would be useful for any purpose. Indeed, the Office action has cited two references that are so unrelated that nothing meaningful could result from any combination of the two references.

Conclusion

Because the cited references, alone or combined, do not describe or suggest the claimed invention, appellant respectfully requests that the rejections of claims 1, 3 and 4 be reversed. Because claims 2, 9 and 11 depend from claim 1, appellant respectfully requests that the rejections of claims 2, 9 and 11 be reversed.

Respectfully submitted,

James Trosino
Registration No. 39,862
Attorney for Appellant

APPENDIX A

CLAIMS

1. A development environment for producing a PostScript® printer description (PPD) text file associated with a printer, the development environment comprising:

a base PPD text file that comprises information regarding the printer, the information including text in a first language, the base PPD text file adapted for a first software platform; and

a PPD generator adapted to import the base PPD text file and to generate therefrom a second PPD text file that includes text translated from the first language to a second language.

2. The development environment of Claim 1, further comprising a build file that describes a product or platform specific set of features of the base PPD text file.

3. A method for generating a PostScript® printer description (PPD) text file associated with a printer, the method comprising:

providing a base PPD text file that comprises information regarding the printer, the information including text in a first language, the base PPD text file adapted for a first software platform;

providing a build file that comprises information as to how the base PPD text file should be edited to provide a second PPD text file that includes text translated from the first language to a second language; and

implementing the build file to generate the second PPD text file.

4. A method for revising a PostScript® printer description (PPD) text file associated with a printer, the method comprising:

providing a base PPD text file that comprises information regarding the printer, the information including text in a first language, the base PPD text file adapted for a first software platform, and a build file that comprises information as to how the base PPD text file should be edited to provide a second PPD text file that includes text translated from the first language to a second language;

modifying the base PPD text file to provide a revised PPD text file; and

implementing the build file to generate a second PPD text file from the revised PPD text file.

9. The development environment of Claim 1, wherein the PPD generator parses the base PPD text file and the build file into text based instructions that may be parsed and assembled by PostScript® printer drivers.

11. The development environment of Claim 1, wherein the PPD generator generates the second PPD file that is adapted for a second software platform.

APPENDIX B

CITED REFERENCES

United States Patent [19]

deSilva

[11] Patent Number: 5,857,064

[45] Date of Patent: Jan. 5, 1999

[54] **SYSTEM FOR IMAGING COMPLEX GRAPHICAL IMAGES**

[75] Inventor: **Mahi deSilva**, Cupertino, Calif.

[73] Assignee: **Object Technology Licensing Corporation**, Cupertino, Calif.

[21] Appl. No.: **946,127**

[22] Filed: **Oct. 7, 1997**

Related U.S. Application Data

[63] Continuation of Ser. No. 415,979, Apr. 3, 1995.

[51] **Int. Cl.**[6] **G06F 15/00**

[52] **U.S. Cl.** **395/112**; 345/522

[58] **Field of Search** 395/101, 104, 395/106, 109, 110, 111, 112, 113, 114, 115, 653; 364/130, 131, 148; 345/501, 502, 503, 506, 522, 523, 526; 382/276, 294, 299, 302, 303; 358/401, 448

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,928,243	5/1990	Hodges	364/519
5,157,765	10/1992	Birk et al.	395/163
5,185,599	2/1993	Doornink et al.	345/200
5,261,029	11/1993	Abi-Ezzi et al.	345/123
5,333,246	7/1994	Nagasaka	395/133
5,502,804	3/1996	Butterfield et al.	395/147
5,524,186	6/1996	Campbell	395/115
5,566,278	10/1996	Patel et al.	395/114

OTHER PUBLICATIONS

Foley et al. "Computer Graphics Principles and Practice", 1990, Addison Wesley, US, p. 923 et al.

Apple Computer, Quickdraw GX Printing, 1994, Addison Wesley, US, pp. 1–3 et al.

Microsoft Corp. "Win32 Programmer's Reference", V.1, 1993, Microsoft Press, US, p. 786 et al.

Primary Examiner—Scott Rogers

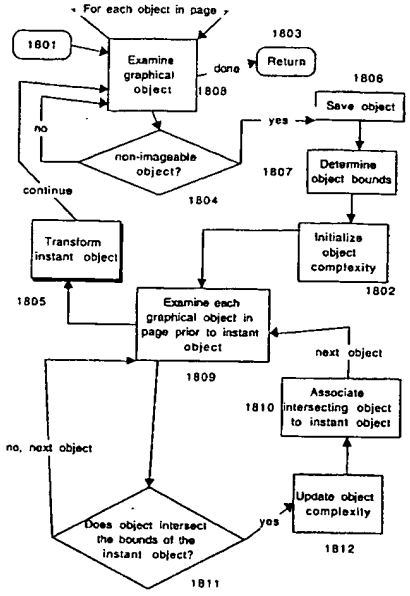
Assistant Examiner—Gabriel I. Garcia

Attorney, Agent, or Firm—Kudirka & Jobse; James A. Ward

[57] **ABSTRACT**

An imaging system that provides a facility for converting highly functional complex graphical primitives that cannot be directly imaged by a less functional page description languages (PDL) such as PostScript or PCL into a plurality of less complex graphical primitives that can be directly imaged by a PDL. Some of these highly functional complex objects are three dimensional objects, high order curves, non regular areas, sampled three dimensional surfaces, off-centered pens, color blending and user programmable three dimensional shaders. These graphical objects are examined by a device handler, the characteristics of the object compared with the capabilities of the PDL of the device and if required the graphical object is transformed into a plurality of simpler objects that can be imaged by the PDL. Depending on the complexity of the graphical primitive, this transformation either creates a pixelmap of the image that is sent to the device at the device's resolution, or the transformation creates a plurality of graphic primitives supported by the device's PDL. Regardless, the desired complex image is printed, and the developer of an application program is able to utilize complex graphical primitives to develop an image that will be presented on a device irrespective of the limitations inherent in the device's actual PDL.

22 Claims, 21 Drawing Sheets



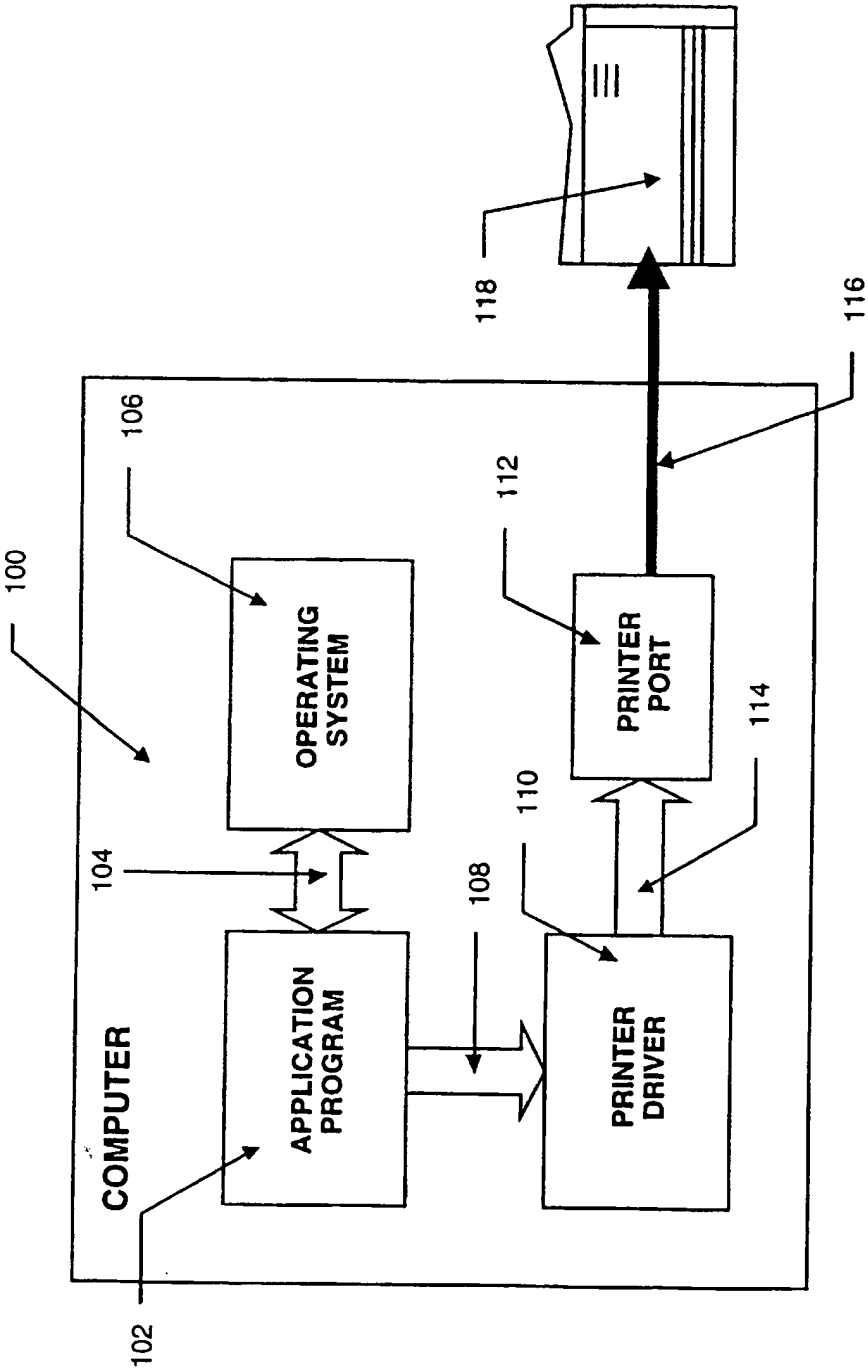


FIG. 1 (Prior Art)

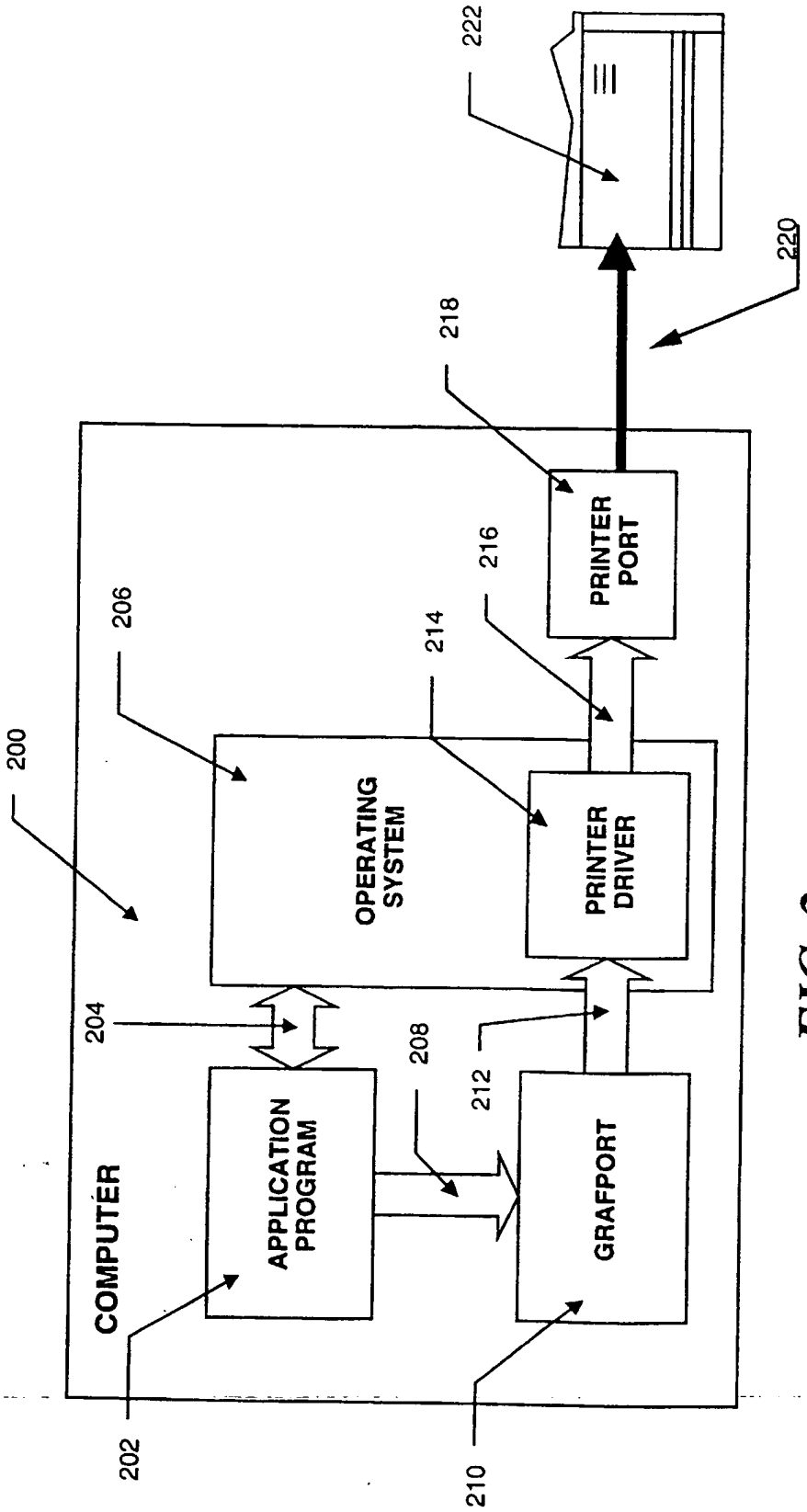


FIG. 2 (Prior Art)

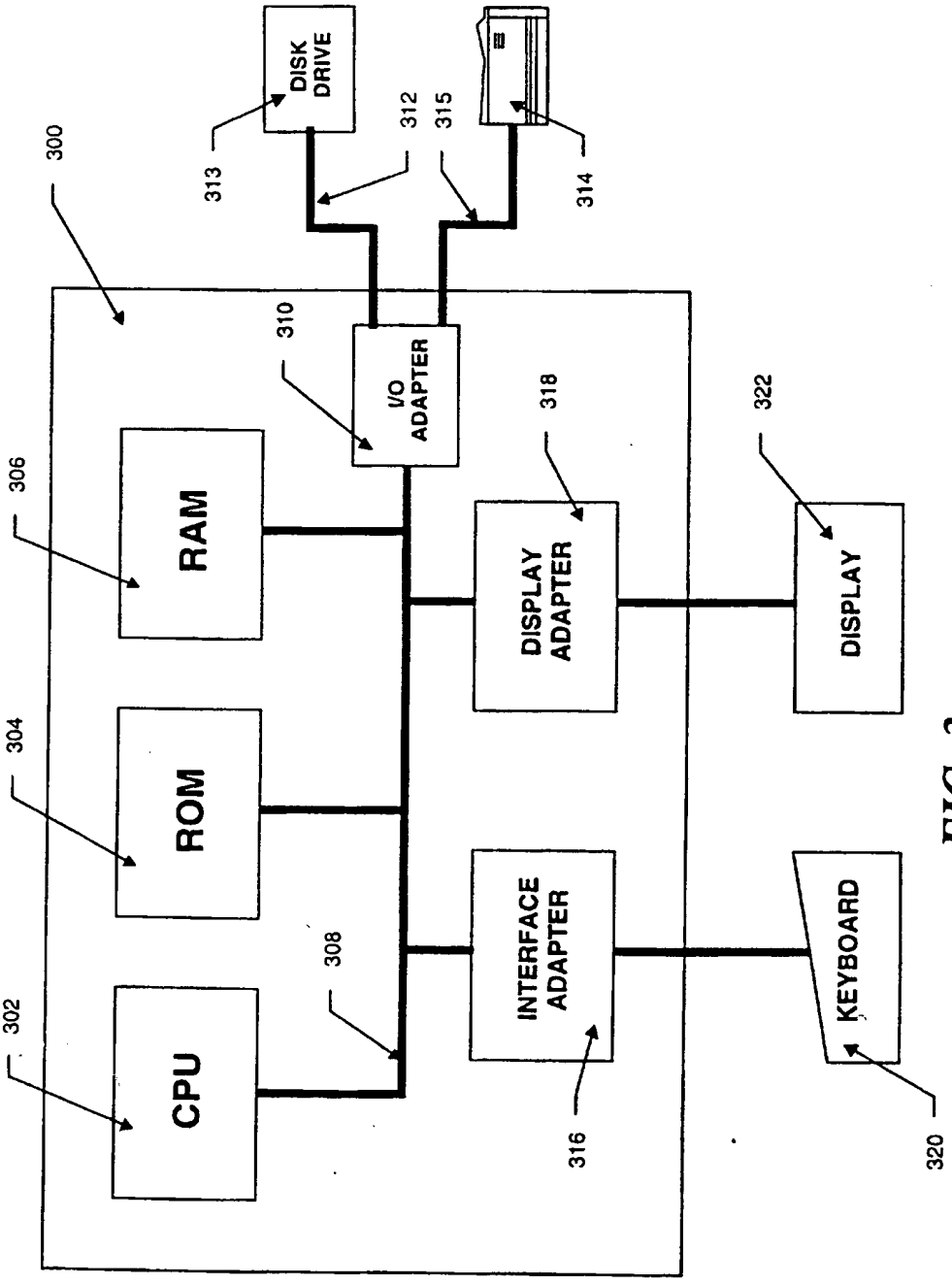

DISK DRIVE
313
312
315
314
300
310
I/O ADAPTER
318
322
306
RAM
DISPLAY ADAPTER
DISPLAY
304
ROM
308
INTERFACE ADAPTER
KEYBOARD
302
CPU
316
320

FIG. 3

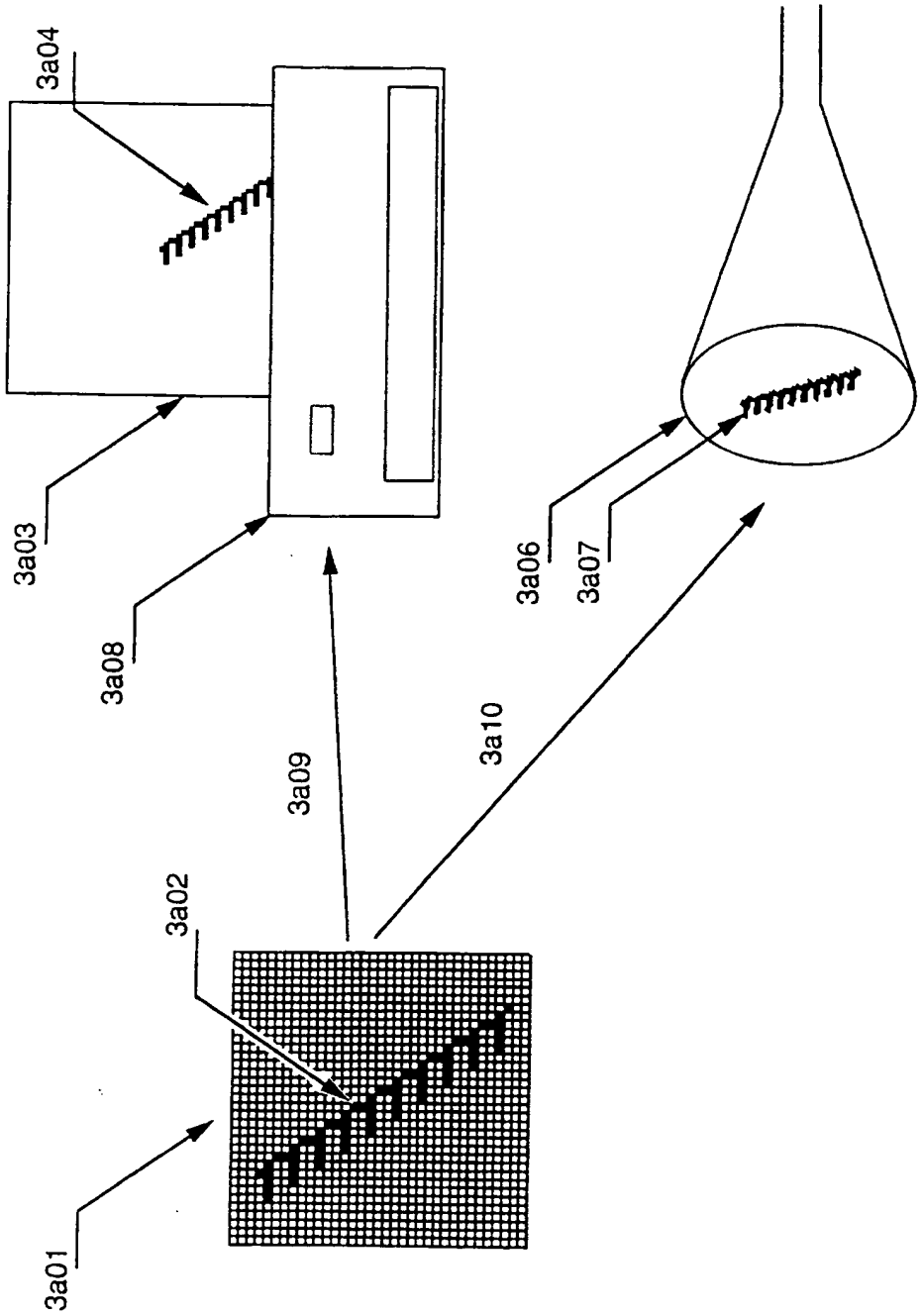


Fig. 3a

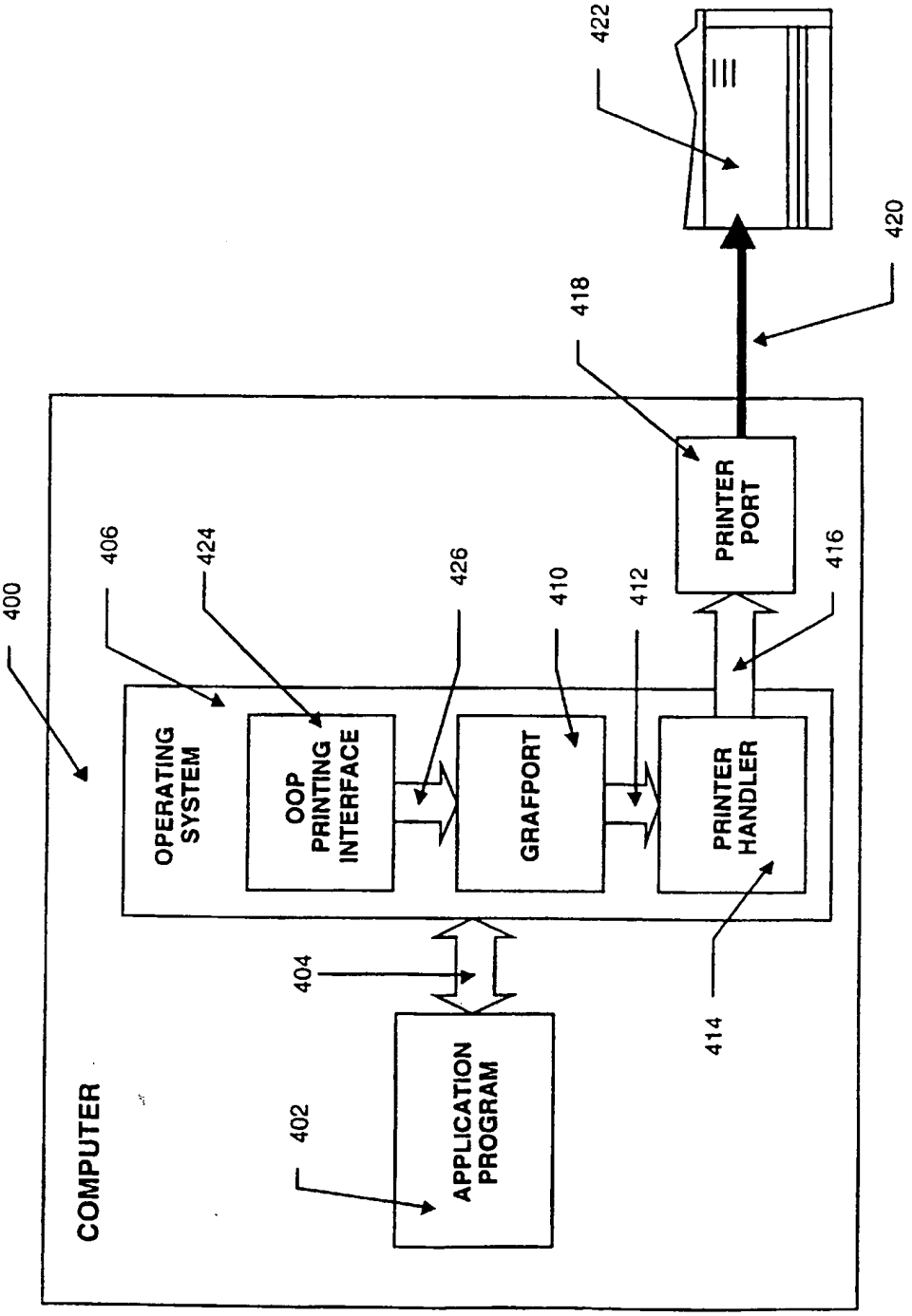


FIG. 4 (Prior Art)

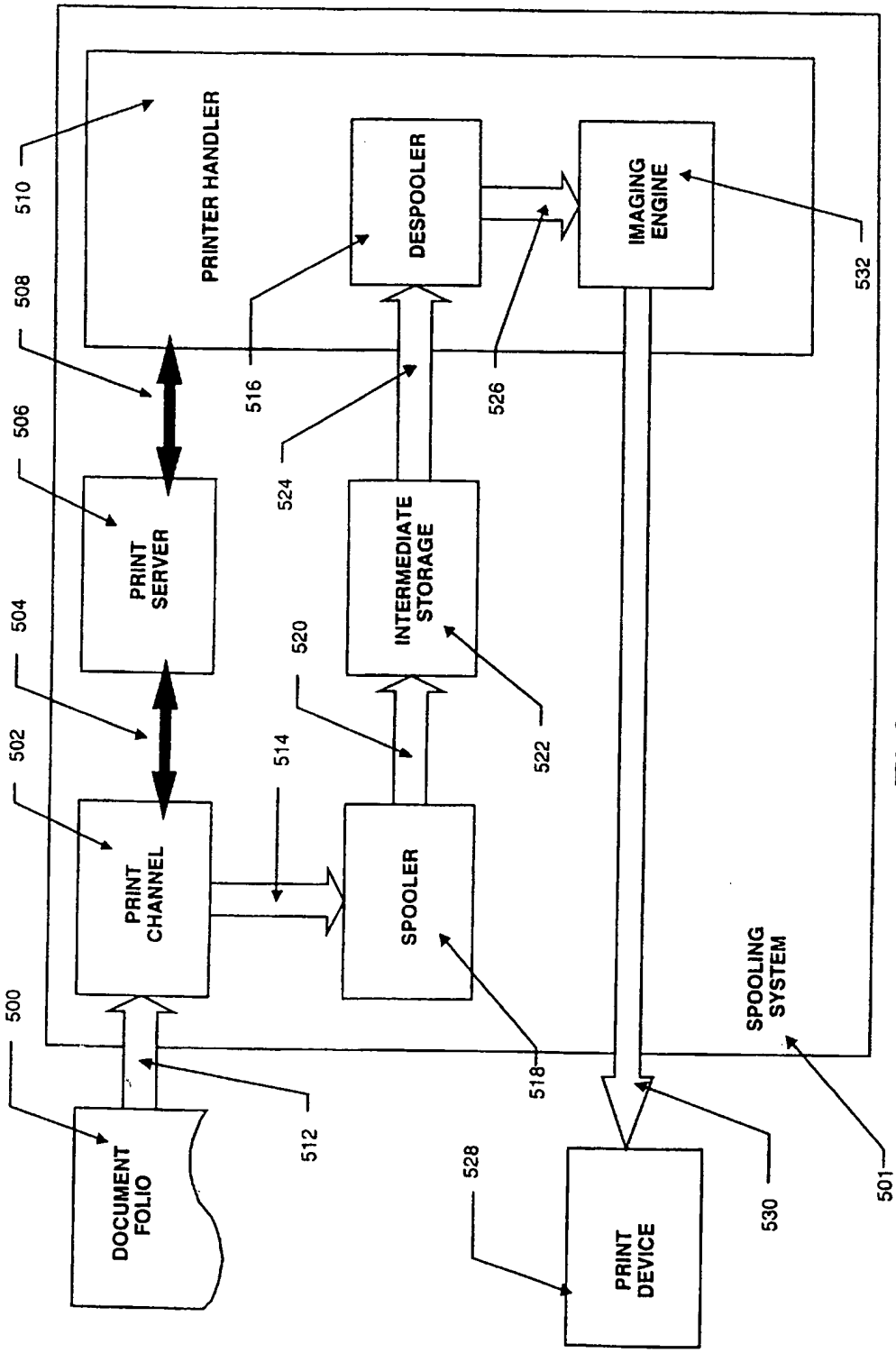

PRINTER HANDLER
DESPOOLER
IMAGING ENGINE
PRINT SERVER
INTERMEDIATE STORAGE
PRINT CHANNEL
SPOOLER
SPOOLING SYSTEM
DOCUMENT FOLIO
PRINT DEVICE
500
501
502
504
506
508
510
512
514
516
518
520
522
524
526
528
530
532

FIG. 5

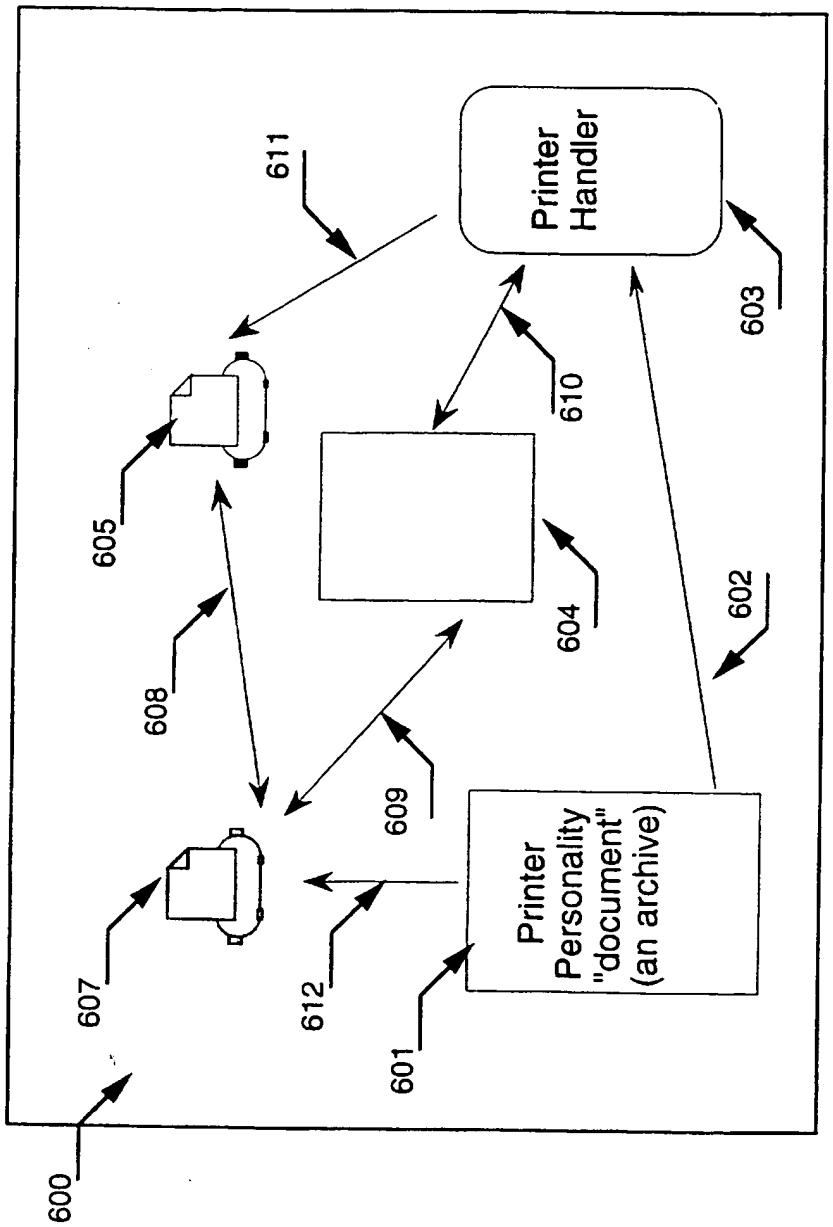

Printer Handler
Printer Personality "document" (an archive)
600
601
602
603
604
605
607
608
609
610
611
612

Fig. 6

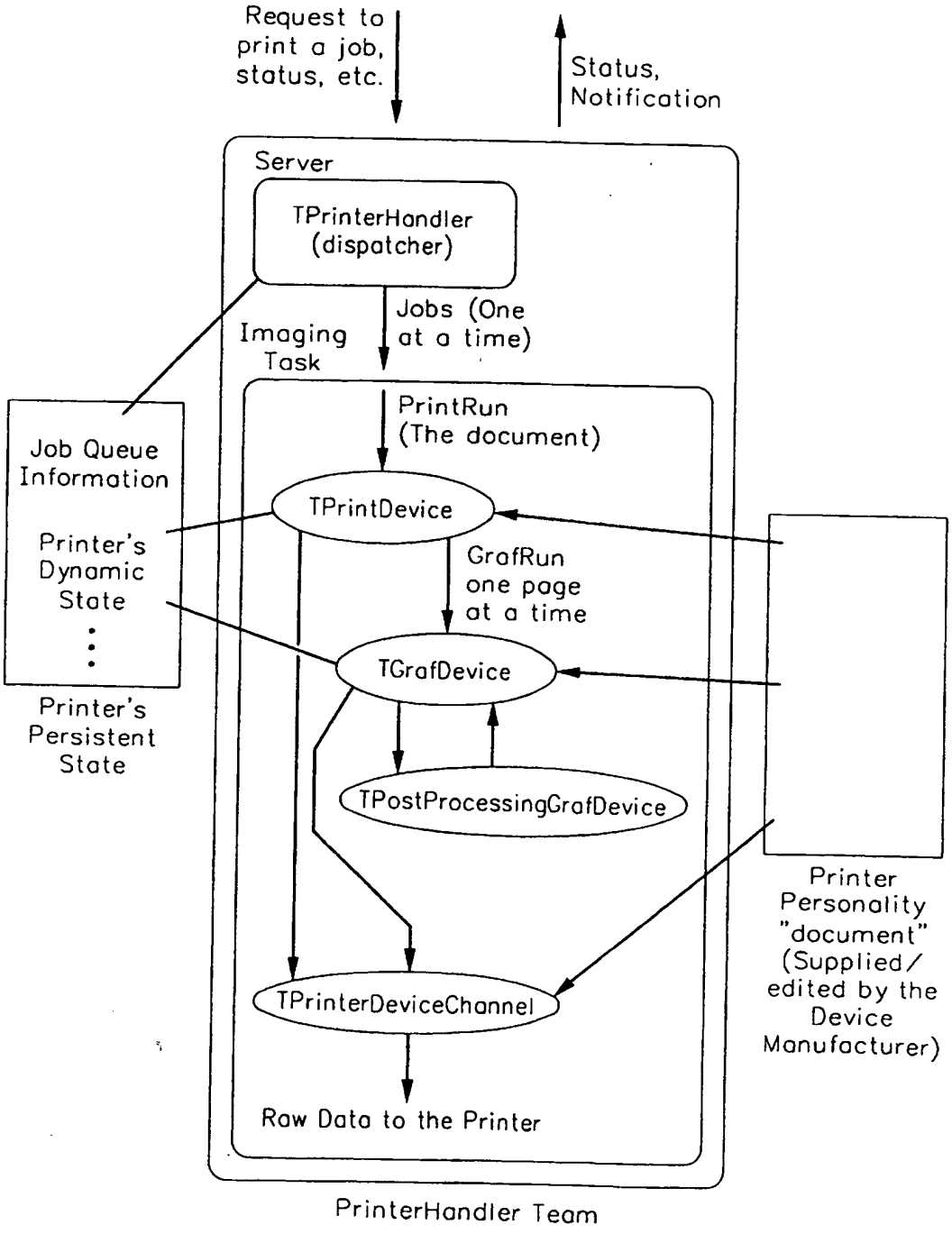

Request to print a job, status, etc.
Status, Notification
Server
TPrinterHandler (dispatcher)
Jobs (One at a time)
Imaging Task
PrintRun (The document)
Job Queue Information
Printer's Dynamic State
TPrintDevice
GrafRun one page at a time
TGrafDevice
Printer's Persistent State
TPostProcessingGrafDevice
TPrinterDeviceChannel
Raw Data to the Printer
Printer Personality "document" (Supplied/ edited by the Device Manufacturer)
PrinterHandler Team

FIG. 7

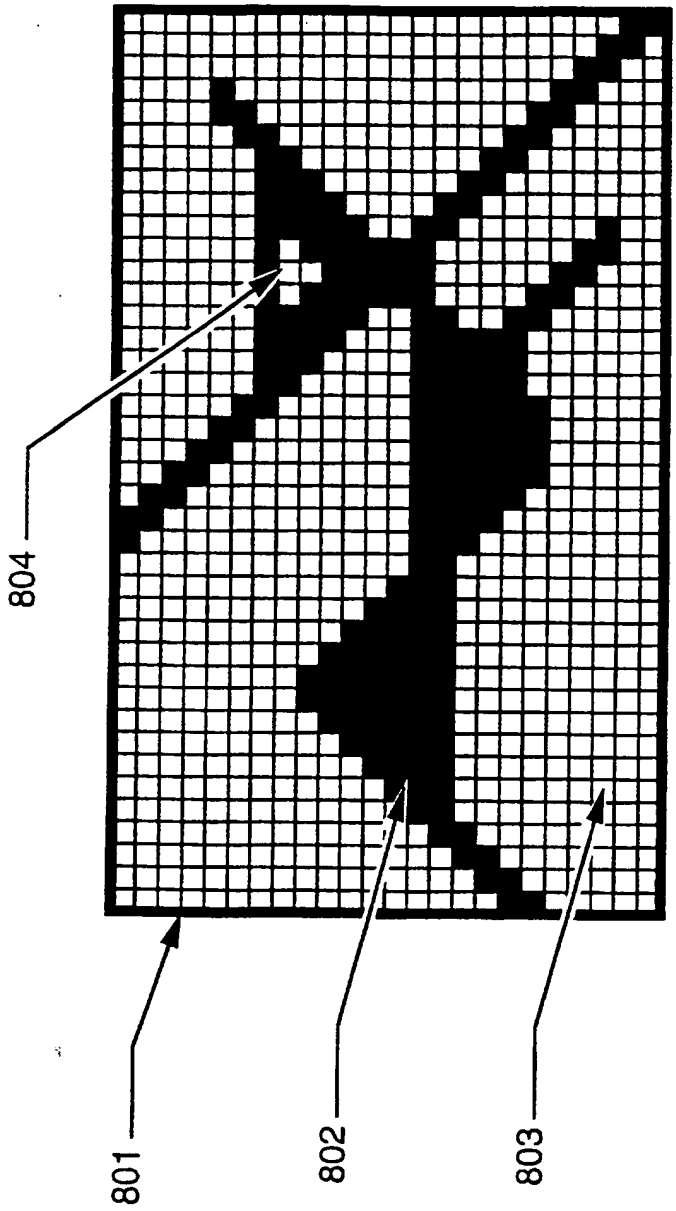


Fig. 8

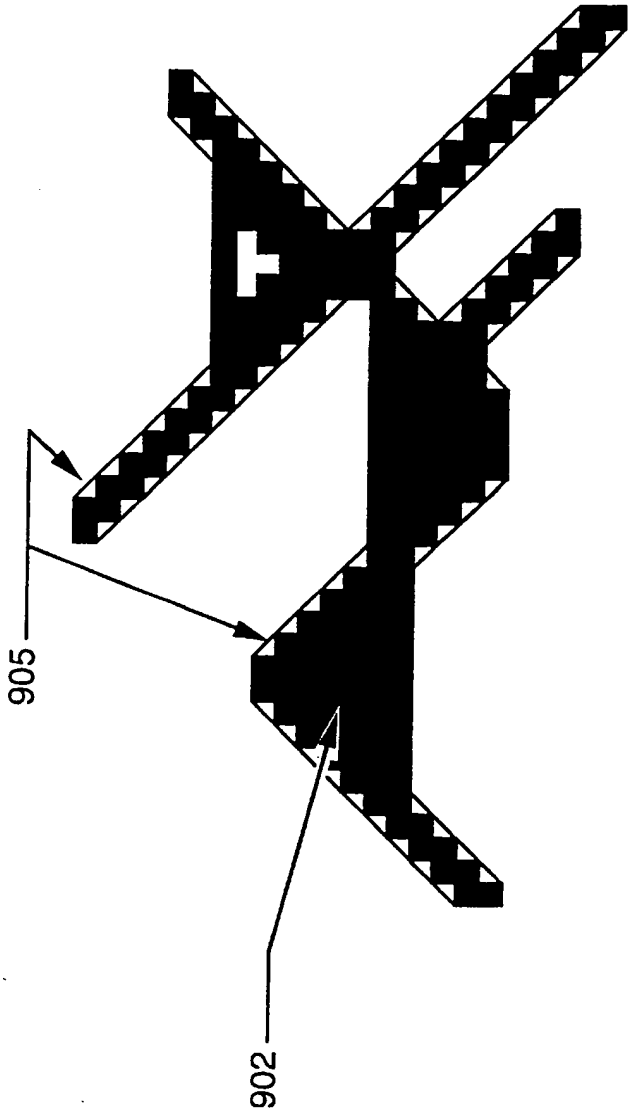


Fig. 9

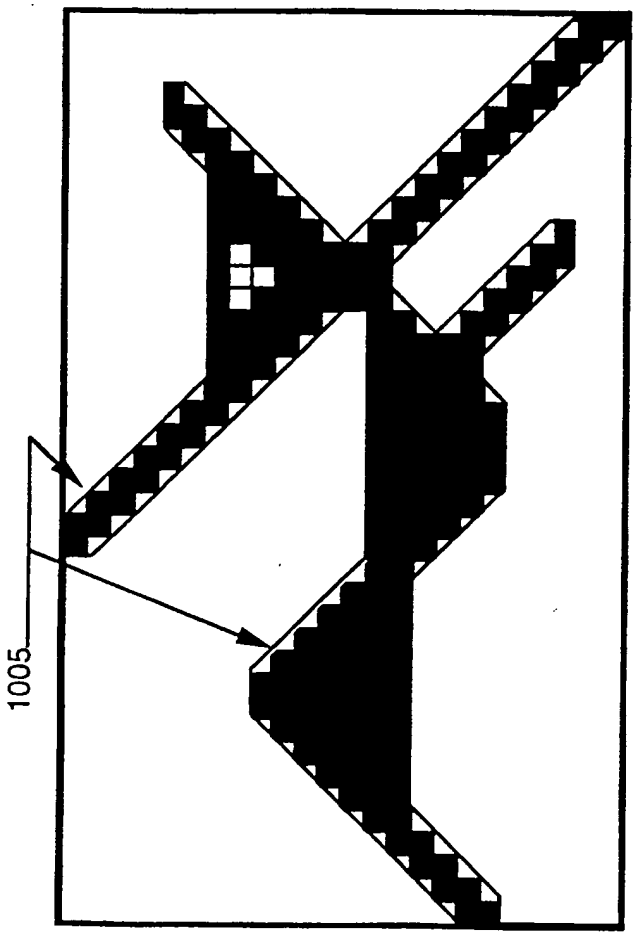


Fig. 10

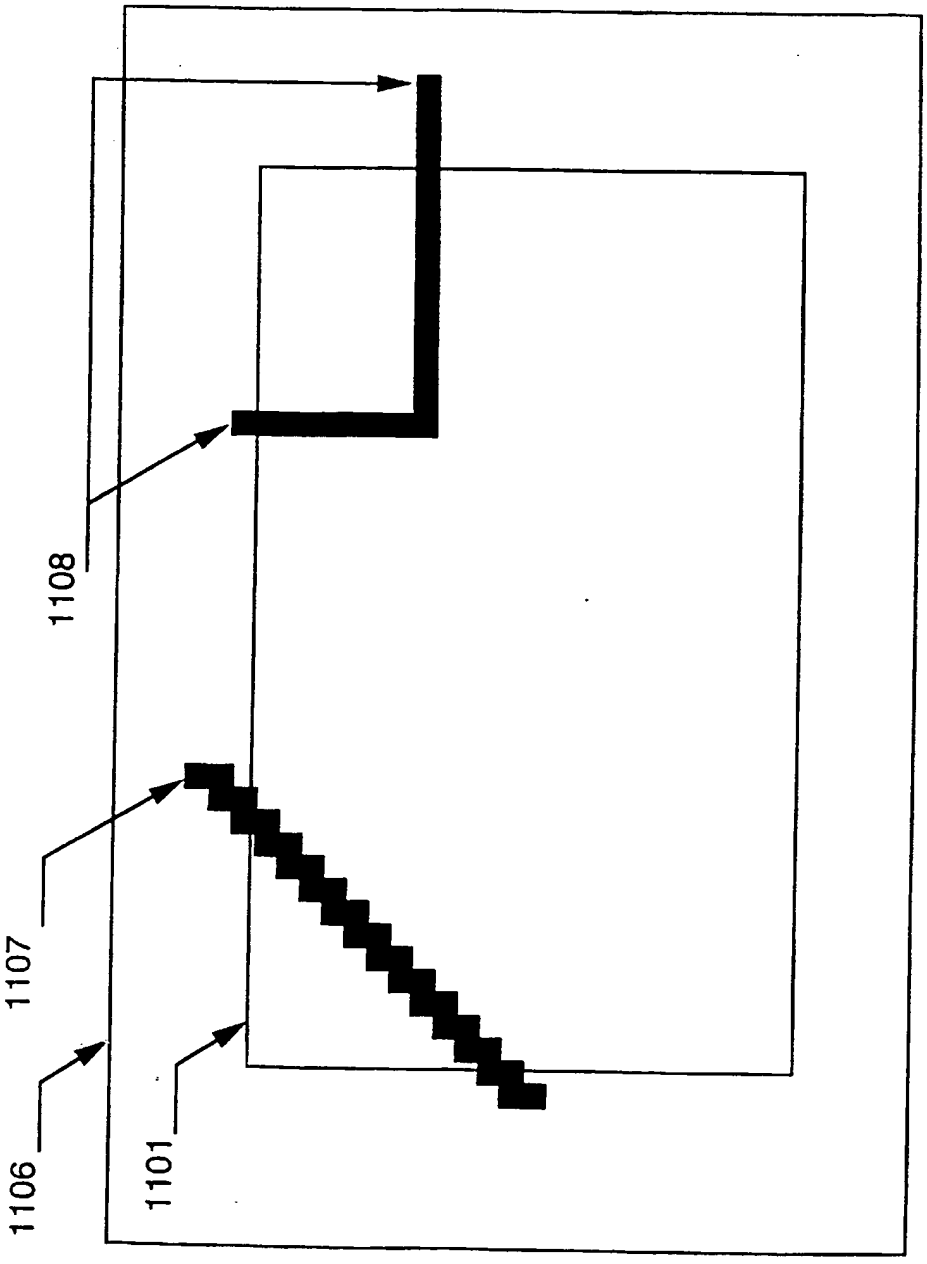


Fig. 11

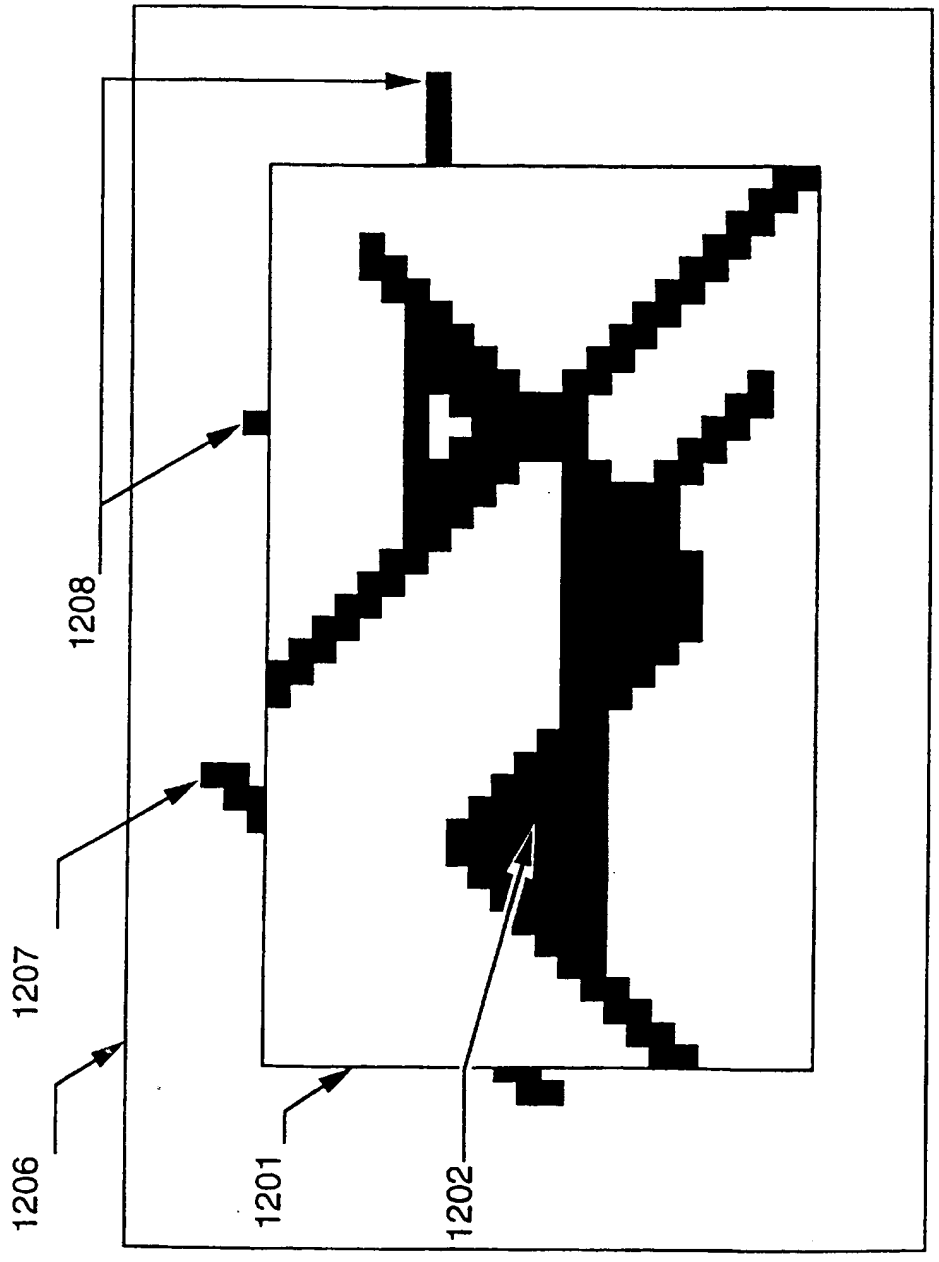


Fig. 12

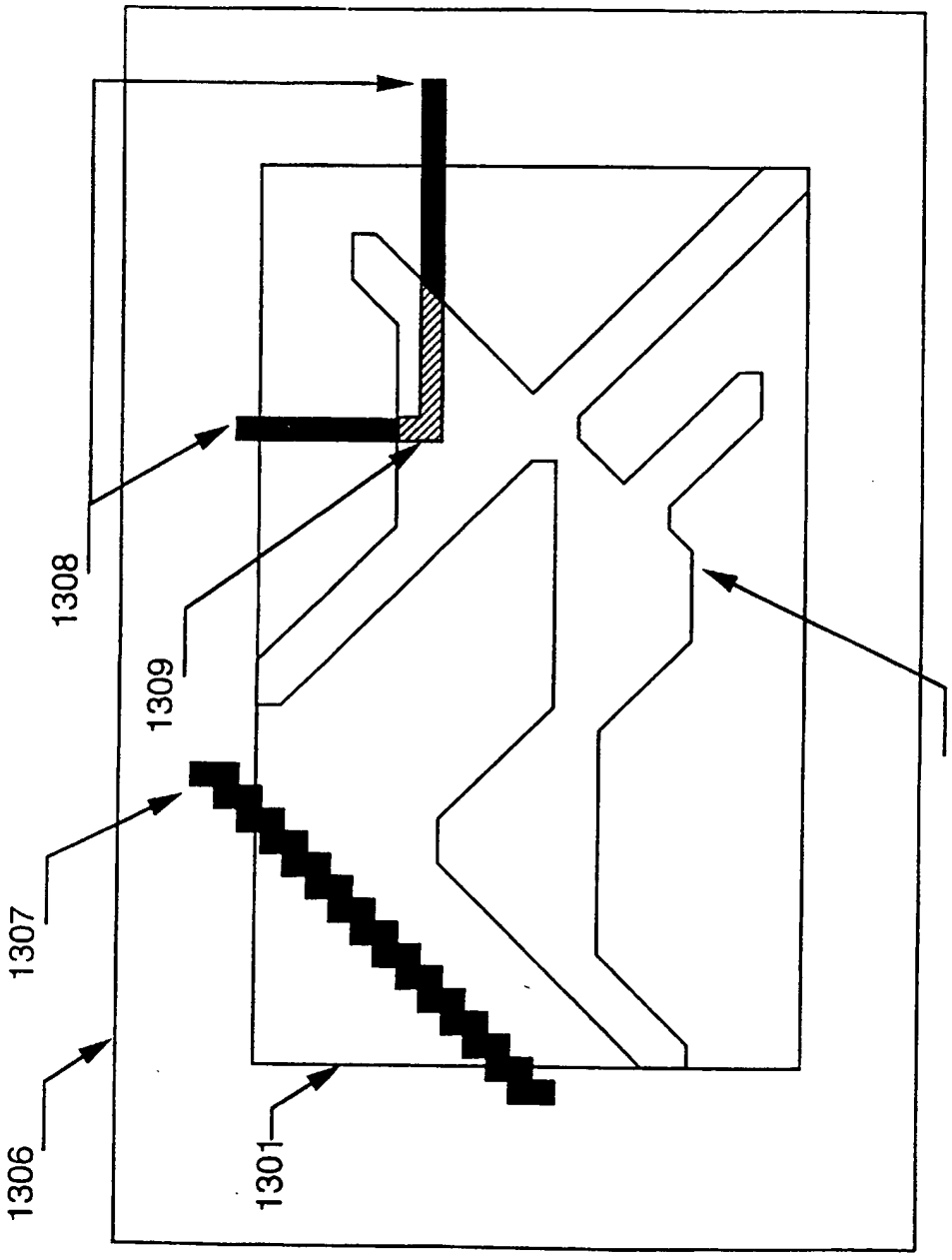


Fig. 13

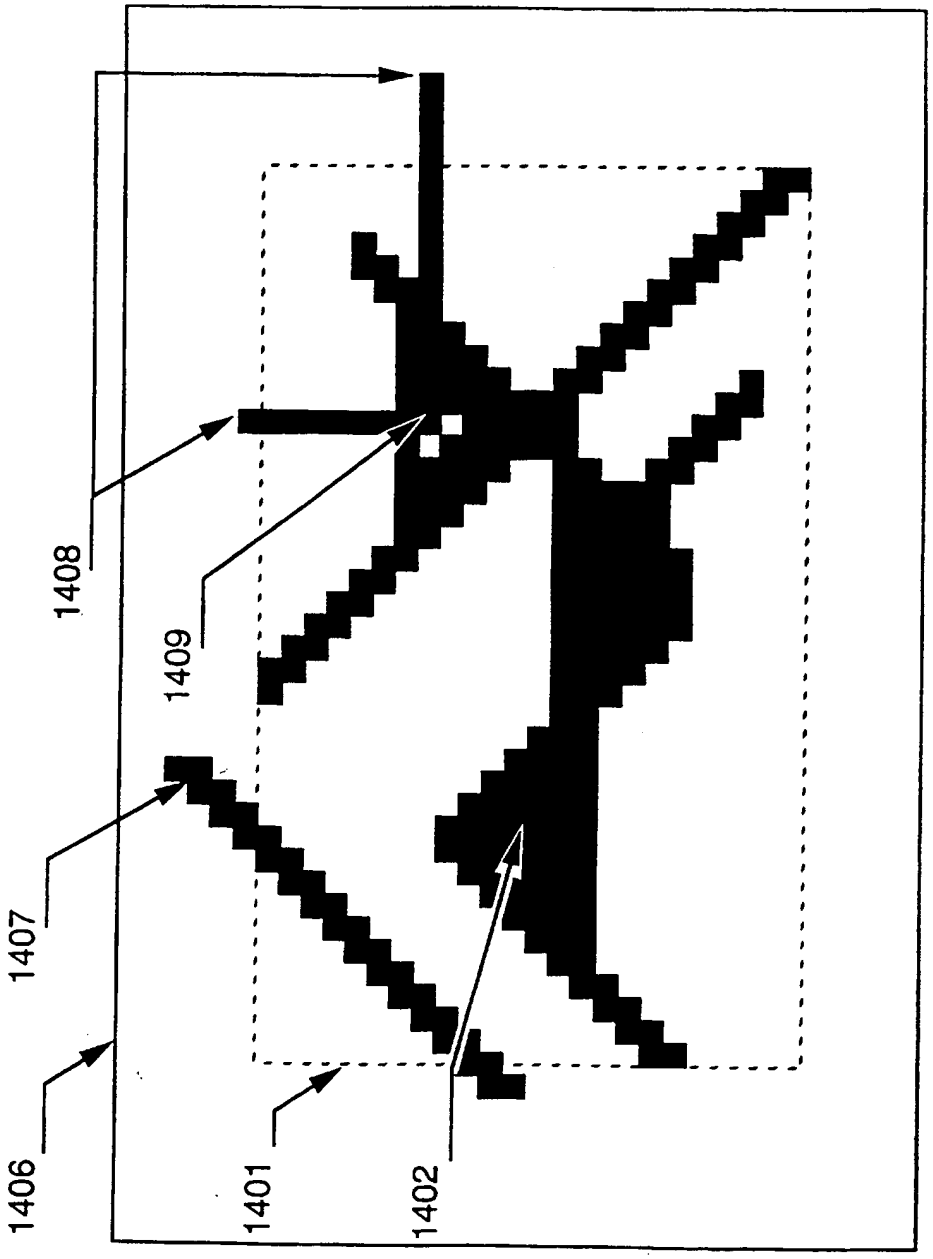


Fig. 14

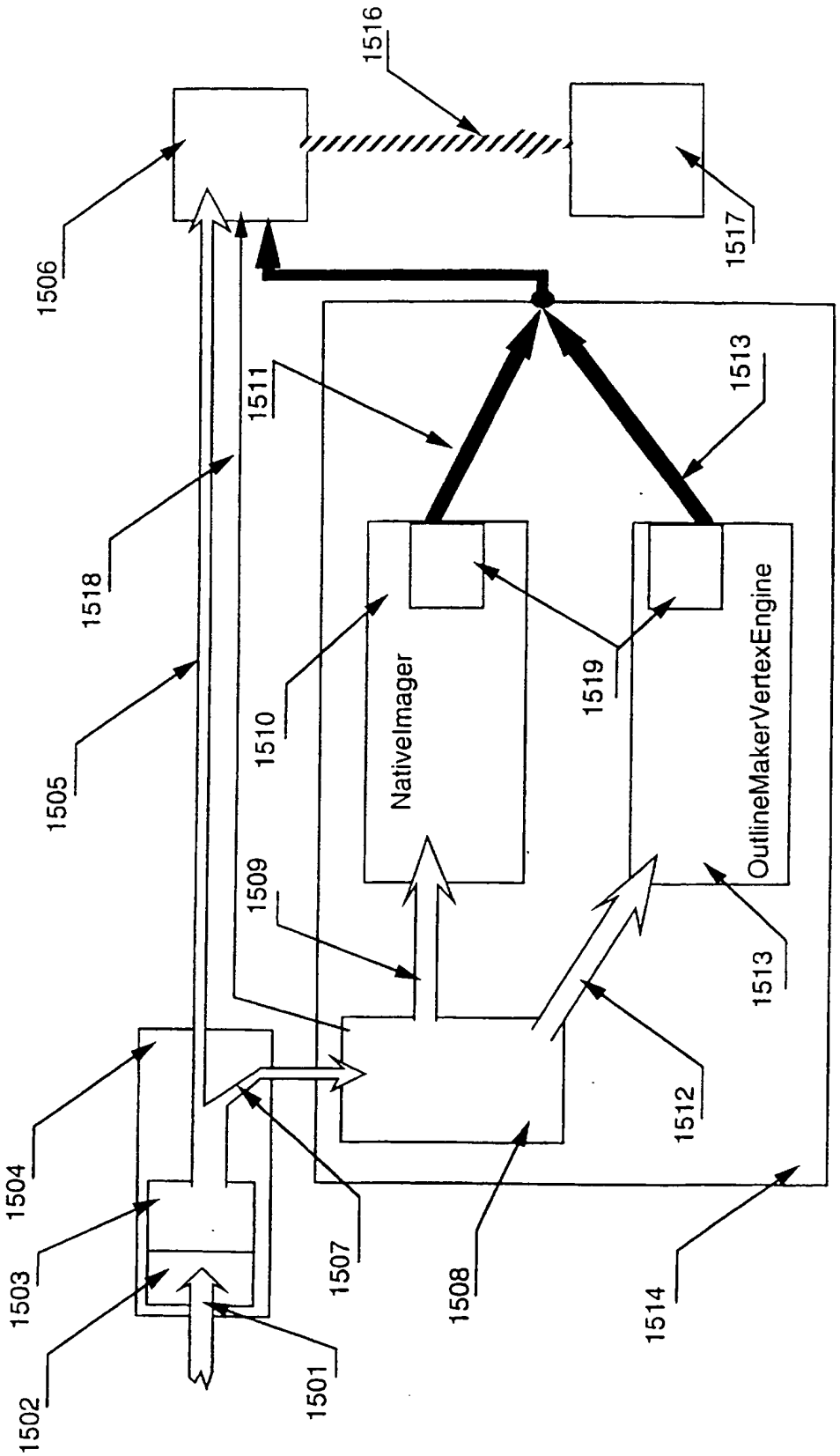

1516
1506
1517
1511
1513
1518
1510
NativeImager
1519
OutlineMakerVertexEngine
1505
1509
1513
1504
1512
1503
1507
1508
1514
1502
1501

Fig. 15

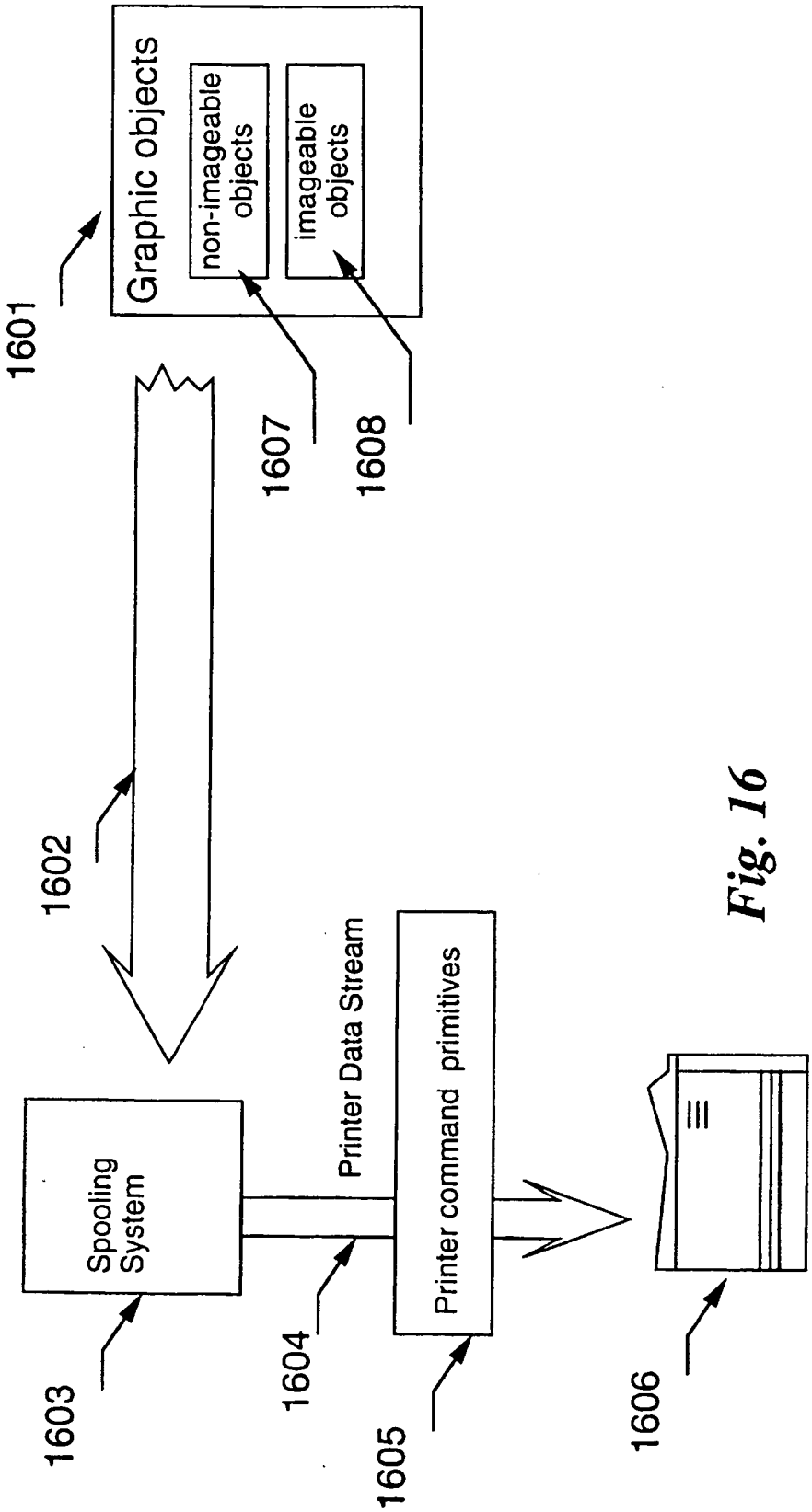

Graphic objects
non-imageable objects
imageable objects
1601
1607
1608
1602
Spooling System
Printer Data Stream
Printer command primitives
1603
1604
1605
1606

Fig. 16

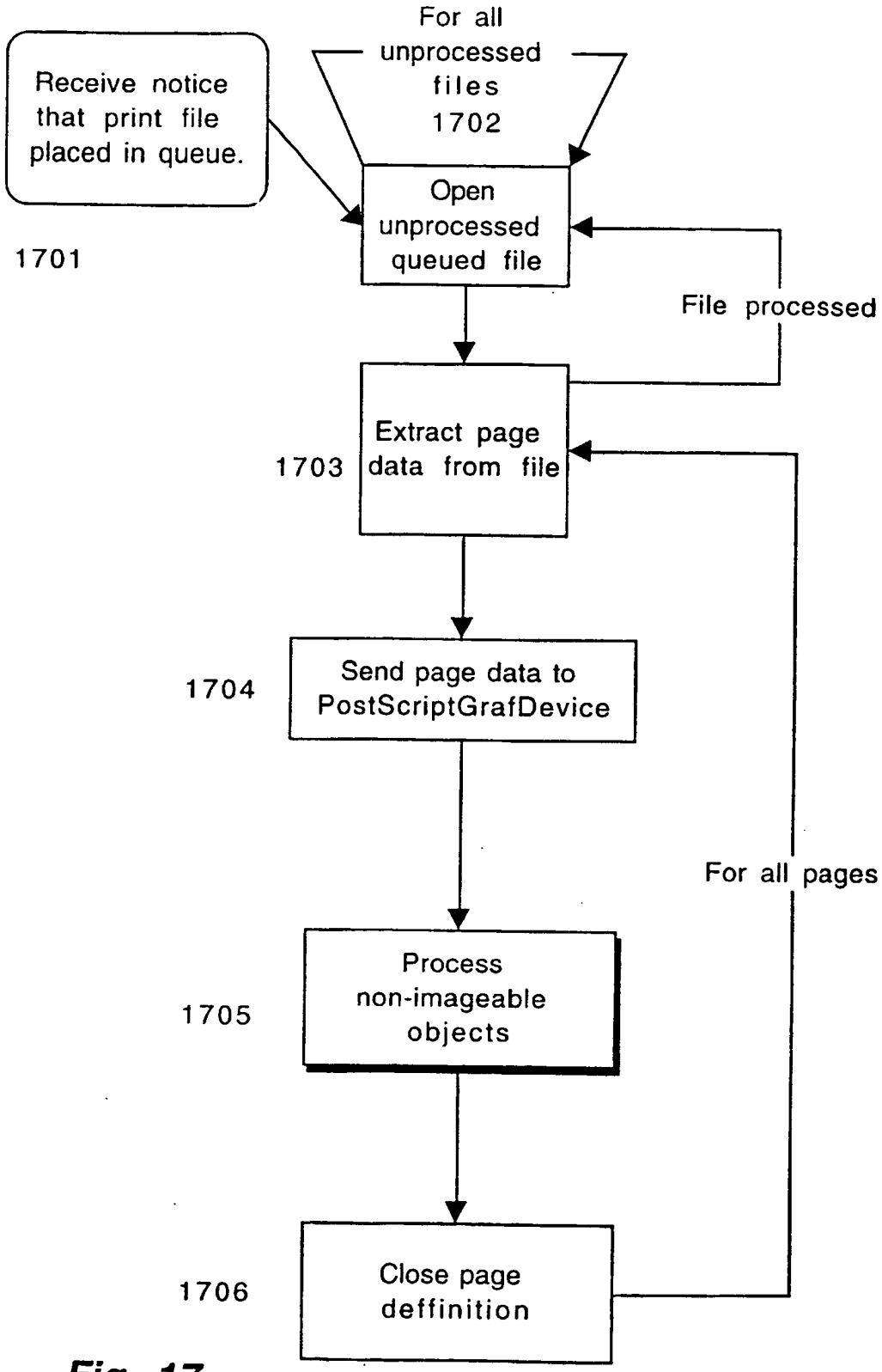

Receive notice that print file placed in queue.
1701
For all unprocessed files 1702
Open unprocessed queued file
File processed
1703
Extract page data from file
1704
Send page data to PostScriptGrafDevice
For all pages
1705
Process non-imageable objects
1706
Close page deffinition

Fig. 17

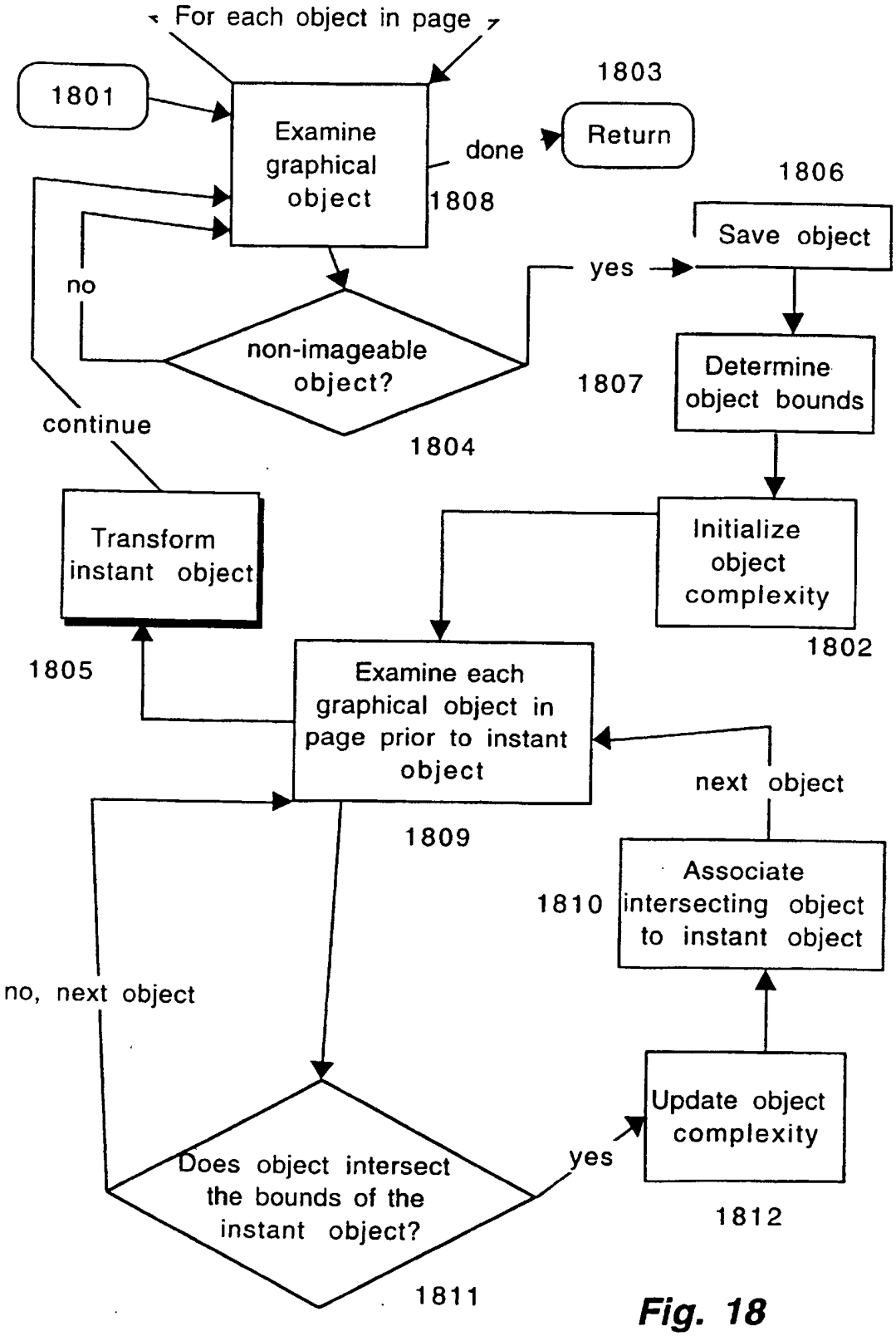

For each object in page
1801
Examine graphical object
1808
1803
Return
done
1806
Save object
yes
no
non-imageable object?
1804
1807
Determine object bounds
continue
Transform instant object
1805
Initialize object complexity
1802
Examine each graphical object in page prior to instant object
1809
next object
Associate intersecting object to instant object
1810
no, next object
Does object intersect the bounds of the instant object?
1811
yes
Update object complexity
1812

Fig. 18

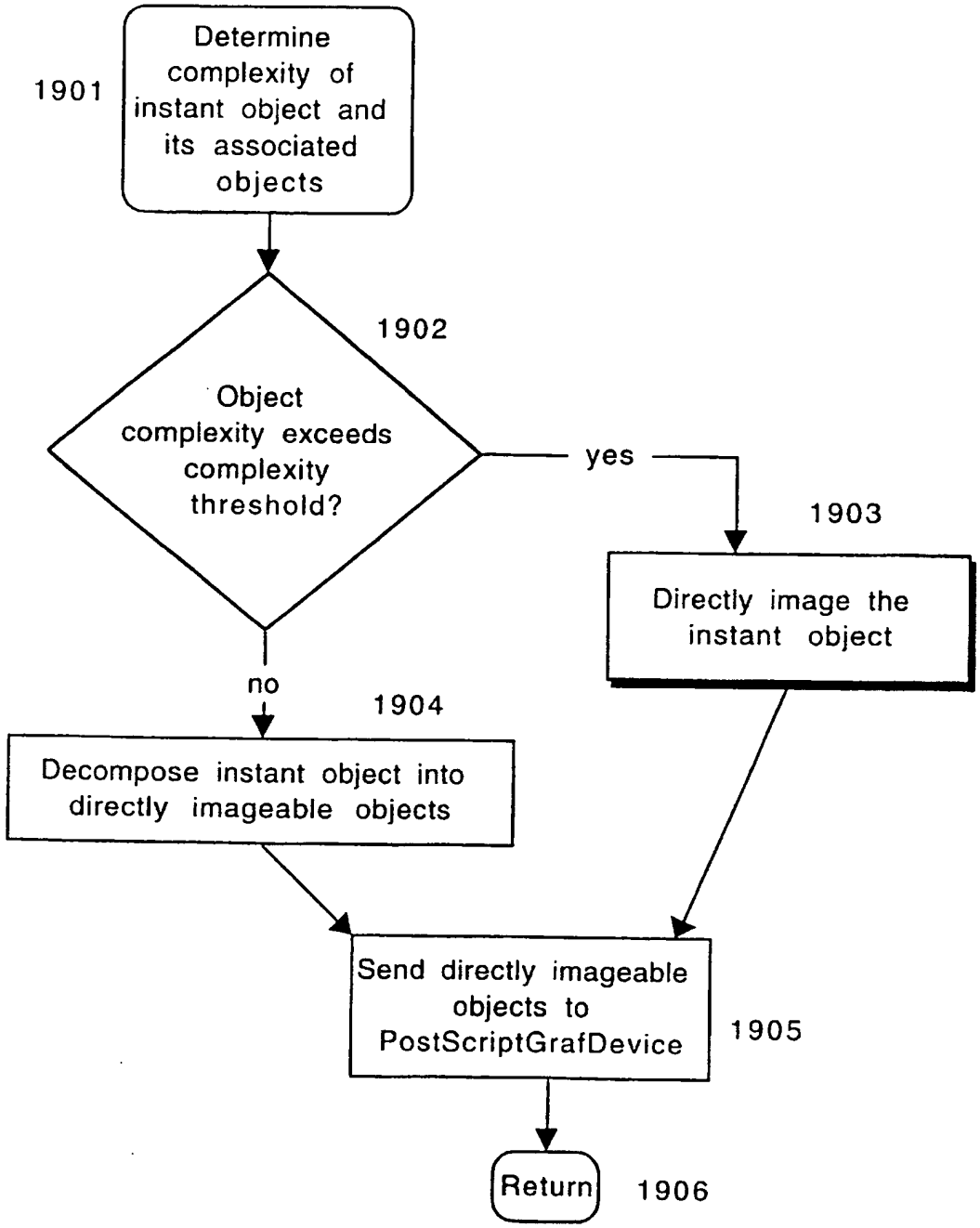

Determine complexity of instant object and its associated objects
1901
1902
Object complexity exceeds complexity threshold?
yes
1903
Directly image the instant object
no
1904
Decompose instant object into directly imageable objects
Send directly imageable objects to PostScriptGrafDevice
1905
Return
1906

Fig. 19

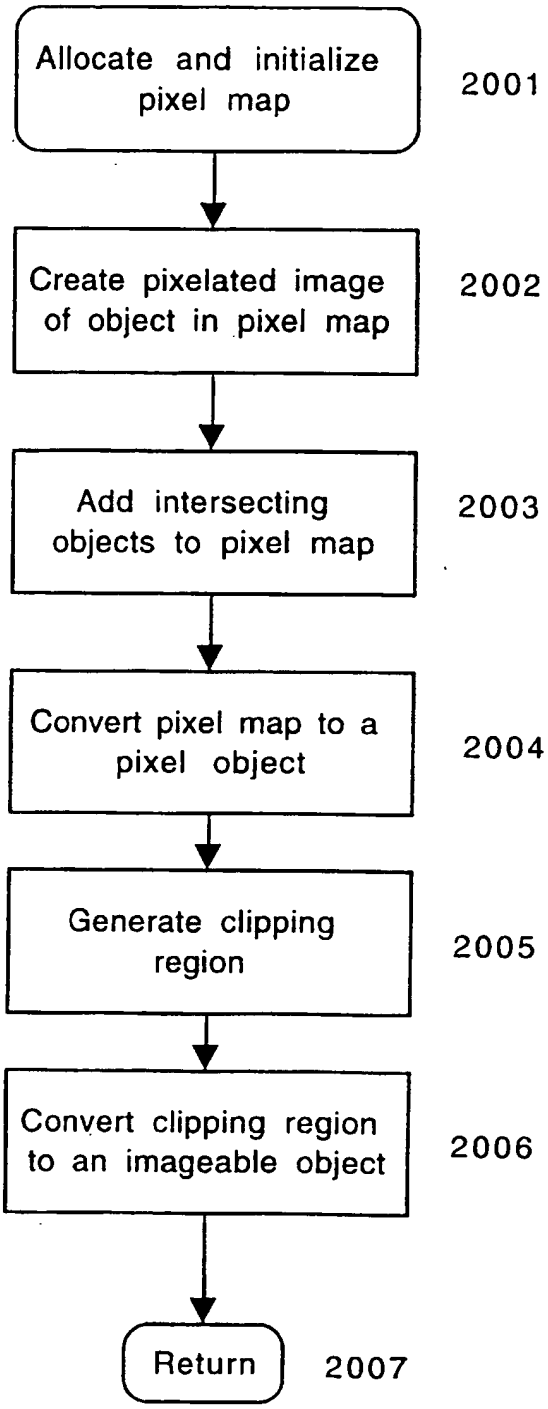

Allocate and initialize pixel map
2001
Create pixelated image of object in pixel map
2002
Add intersecting objects to pixel map
2003
Convert pixel map to a pixel object
2004
Generate clipping region
2005
Convert clipping region to an imageable object
2006
Return
2007

Fig. 20

SYSTEM FOR IMAGING COMPLEX GRAPHICAL IMAGES

This is a continuation of application Ser. No. 08/415,979, filed Apr. 3, 1995.

COPYRIGHT NOTIFICATION

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

This invention generally relates to improvements in computer systems and, more particularly, to operating system software for displaying and printing documents.

BACKGROUND OF THE INVENTION

One of the most important aspects of a modem computer system is the ability to generate an image of textual or graphical information. This information can be manipulated by the computer, visually displayed and stored. This display is generally performed by a display monitor and by a "hard" printed copy. To create the hard copy of this information, a computer system generally includes a printing device which is electrically connected to the computer system and controlled by it in order to generate a permanent image on a selected medium. Examples of raster oriented printing devices in common use are dot matrix, ink jet and LASER printers that fix permanent images on paper under control of the computer. Although paper is the most common medium, other media are often used, such as photographic film and overhead transparencies.

FIG. 3*a* indicates the correspondence between a computer memory raster **3*a*01** contains an image **3*a*02** which can be displayed on a raster monitor **3*a*06** or a raster printer **3*a*08**. The memory raster **3*a*01** is sent to the printer **3*a*08** as indicated by arrow **3*a*09**. The printer **3*a*08** transfers the raster image **3*a*02** to a resulting image **3*a*04** on paper **3*a*03**. Similarly, the memory raster **3*a*01** is sent to a monitor **3*a*06** as indicated by arrow **3*a*10**. The display **3*a*06** transfers the raster image **3*a*02** to a resulting image **3*a*07** on the display **3*a*06**. Images **3*a*07** and **3*a*04** are dependent on the resolution of each device. The same image **3*a*02** if sent to devices of differing resolution will be produced at different sizes.

In order to print a document that is displayed on the monitor or stored within the memory, several actions must take place. First, since the print medium generally has a fixed size, the printable information must be divided into pieces which are small enough to fit on the selected medium, a process which is called pagination. In addition, the information may need to be reformatted from the format in which it is either displayed or stored into a format which is suitable for controlling the printing device to actually perform the printing on the medium. The reformatting in this latter step may include the insertion of control commands into the printable information in order to control the printing device. These added commands may, for example, include such commands as carriage returns, line feeds, form feeds, spaces and font information, all of which format the printable information. The reformatting may also include a conversion step in which the information is converted into the form

used by the printing device. For example, if the printer uses the PostScript page description language, this reformatting would include the creation of a PostScript program to generate the printed page. Similarly, if the printer uses the PCL page description language.

The pagination and reformatting necessary to convert the printable information into a form which can be printed on a given printing device is generally performed by software programs running within the computer system. Software programs operating on a computing system generally can be categorized into two broad classes: operating systems which are usually specific to a type of computer system and consist of a collection of utility programs that enable the computer system to perform basic operations, such as storing and retrieving information on a peripheral disk memory, displaying information on an associated video display, performing rudimentary file operations including the creation, naming and renaming of files and, in some cases, performing diagnostic operations in order to discover or recover from malfunctions.

By itself, the operating system generally provides only very basic functions and must be accompanied by an "application" program. The application program interacts with the operating system to provide much higher level functionality and a direct interface with the user. During the interaction between the application program and the operating system, the application program typically makes use of operating system functions by sending a series of task commands to the operating system which then performs the requested tasks. For example, the application program may request that the operating system store particular information on the computer disk memory or display information on the video display.

FIG. **1** is a schematic illustration of a typical computer system utilizing both an application program and an operating system. The computer system is schematically represented by box **100**, the application program is represented by box **102** and the operating system by box **106**. The interaction between the application program **102** and the operating system **106** is illustrated schematically by arrow **104**. This dual program system is used on many types of computers systems ranging from mainframes to personal computers.

The method for handling printing, however, varies from computer to computer, and, in this regard, FIG. **1** represents a prior art personal computer system. In order to provide printing functions, the application program **102** interacts (as shown schematically by arrow **108**) with printer driver software **110**. Printer driver software **110** is generally associated with an application program and reformats and converts the printable information as necessary. Since each printer has its own particular format and control command set, which must be embedded in the text properly to control the printer, the printer driver software **110** must be specifically designed to operate with one printer or one type of printer. Application support for differing types of printers generally requires support for a specialized printer driver for each type of printer supported. Often a multitude of printer drivers were provided with the application to support a wide range of printing devices.

The printer driver software **110** produces a reformatted information stream containing the embedded commands as shown schematically as arrow **114**. The converted information stream is, in turn, applied to a printer port **112** which contains circuitry that converts the incoming information stream into electrical signals. The signals are, in turn, communicated to the printer **118** via some data pathway **116**

such as a dedicated cable or a network. Printer **118** usually contains an "imaging engine" which is a hardware device or a ROM-programmed computer which takes the incoming information stream and converts it into the electrical signals necessary to drive the actual printing elements. The result is a "hard copy" output on the selected medium.

While the configuration shown in FIG. **1** generally works in a satisfactory manner, it has several drawbacks. Since the printer driver software **110** is specific to each type of printer, a separate driver had to be provided for each printer type with which the application program is to operate. In the personal computer market, there are a large number of different printer types that are compatible with each type of computer and, therefore, as the number of printer types proliferated, so did the number of printer drivers which were required for each application program so that the program was compatible with most available printers. Therefore, application program developers had to provide, test, and support larger and larger numbers of printer drivers with each application program, resulting in wasted time and effort and wasted disk space to hold the drivers, only one or two of which were of interest to any particular user. Unfortunately, if a user purchased an application program and it did not include a printer driver which could control the printer which the user owned, unpredictable operation occurred, resulting in program returns and user dissatisfaction.

In addition, it was also necessary for each application program to provide high level printing functions such as pagination and page composition (including addition of margins, footnotes, figure numbers, etc.) if such functions were desired. Consequently, each application program developer had to spend time developing programs to implement common printing functions which programs were specific to each application program, thereby resulting in duplicated effort and wasted time.

In order to overcome the aforementioned difficulties, the prior art arrangement was modified as shown in FIG. **2**. In this new arrangement, computer system **200** is still controlled by application program **202** which cooperates, as shown schematically by arrow **204**, with operating system **206**. However, in the system shown in FIG. **2** operating system **206** includes printer drivers **214**. A separate printer driver must still be provided for each different type of printer, but the printer drivers are sold with, and part of, the operating system. Consequently, it is not necessary for each application program to have its own set of printer drivers. An application program, such as application program **202**, communicates with the printer driver **214** by means of a standardized interface **210**. Two common interfaces are called "grafports" or "device contexts". Illustratively, application program **202** provides information (schematically shown by arrow **208**) in a standardized form to the grafport **210**. The grafport **210**, in turn, forwards information, as shown by arrow **212**, to printer driver **214** which reformats and converts the information as previously described into the format required by the printer. The output of printer driver **214** is provided (illustratively shown as arrow **216**) to printer port **218** where it is converted to electrical signals that are transmitted, via a data pathway **220**, to the printer **222**.

The configuration shown in FIG. **2** has the advantage that the application program developer need not worry about the specific computer and printer combination on which the program will ultimately run in order to provide printing capabilities to the application program. However, it still suffers from the drawback that, if desired, high level printing capabilities such as pagination and page composition must still be designed into each application program, thereby duplicating program code and wasting programming resources.

These problems were addressed by providing an object-oriented printing interface that communicated with the remainder of the operating system through a standard interface such as a grafport and printer drivers provided for each printer type within the operating system. So that an application not only need not worry about the particular printer/computer combination with which it is to operate, but also need not have a built in document formatting capability. This printing interface included objects that provided queries for device identification, optimized imaging, and printer status. Other objects were also provided for data transfer to bracket connections prior to sending and receiving information. Still other objects were provided for canceling a print job, pausing a job, and clearing out a job. Finally, an object was also provided for supporting multiple streams of communication to an imaging task.

FIG. **4** shows a schematic overview of another prior art computer system utilizing an object-oriented printing interface. The computer system is shown generally as a box **400**, and an application program **402** and an operating system **406** are provided to control and coordinate the operations of the computer. Application program **402** communicates with operating system **406** as indicated by arrow **404**. However, in accordance with an embodiment of the invention, rather than communicating directly with a standard interface, such as grafport **410**, application program **402** can now communicate with operating system **406** at a higher level when it wishes to print information. This latter interaction is accomplished by providing an object-oriented printing interface shown schematically as box **424**. Printing interface **424** responds to a series of simple commands generated by application program **402** in order to perform various formatting and pagination functions. The formatted, printable information is then transmitted to a grafport **410** as indicated schematically by arrow **426**. It is possible for application program **402** to communicate directly with grafport **410** as in the prior art arrangement shown in FIG. **2**, however, it is not expected that this will be the normal mode of operation.

In any case, the information flows through grafport **410**, and as indicated by arrow **412**, is provided to a printer handler **414**. Printer handler **414** is similar to printer drivers previously described. However, it is "intelligent" and offers some additional capabilities. Essentially, printer handler **414** processes the incoming data stream indicated by arrow **412** and adds the necessary printer commands to control the associated printer schematically illustrated as printer **422**. The reformatted printer data stream is provided, as indicated by arrow **416**, to a printer port **418** which converts the data into electrical signals that are sent over data pathway **420** to printer **422**.

Although a significant improvement over the previously described art shown in FIG. **2**, this approach still suffers from the drawback that once the Printing interface **424** provided support for highly functional graphic command objects—objects that could not be simply represented within the printer's PDL—the print driver would either simply not perform the command, or fall back to printing a low resolution pixelmap representative of the image associated with the object. This problem is demonstrated with graphic primitives such as high order curves, sampled 3D surfaces, non-centered pens, color masks and color blending transfer modes, and user programmable 3D shaders. These graphic commands are not supported in either PCL version 5 or PostScript.

SUMMARY OF THE INVENTION

The foregoing problems are overcome and the foregoing object is achieved in an illustrative embodiment of the invention in which an object-oriented printing interface includes a method for transforming graphical objects that are too complex to be directly translated into the printing device's command primitives. The graphical objects are first sent to the printing device where the translatable objects are translated into the device's command primitives. Then the graphical objects are examined to select the objects that require special handling. These objects are examined and based upon the complexity of the object and the capabilities of the printing device, the object is either decomposed into a plurality of imageable objects and printed; or a pixel map of the object is generated, adjusted for intersections with other objects and printed.

brief DESCRIPTION OF THE DRAWINGS

The above and further advantages of the invention may be better understood by referring to the following description in conjunction with the accompanying drawings, in which:

FIG. **1** is a schematic block diagram of a prior art computer system showing the relationship of the application program to the operating system and the printer driver in the prior art;

FIG. **2** is a schematic block diagram of a modification of the prior art system shown in FIG. **1** to allow the application program to interface to a standard printing interface in the prior art;

FIG. **3** is a block schematic diagram of a computer system, for example, a personal computer system on which the inventive object-oriented printing interface operates in accordance with a preferred embodiment;

FIG. **3***a* is a schematic diagram of the correspondence of pixels in memory with the resulting output from a pixel oriented imaging device;

FIG. **4** is a schematic block diagram of a modification of the prior art system shown in FIG. **2** in which the operating system includes an object-oriented printing interface in accordance with a preferred embodiment;

FIG. **5** is a block schematic diagram of the information paths and hardware by which printable information is channeled through intermediate storage to a print device in accordance with a preferred embodiment;

FIG. **6** shows how the personality document is used by different parts of the printing system in accordance with a preferred embodiment;

FIG. **7** details the printer handler components and their interactions in accordance with a preferred embodiment;

FIG. **8** demonstrates how an image is rendered in a pixel map form in accordance with a preferred embodiment;

FIG. **9** shows how a clipping region is used to outline an image in accordance with a preferred embodiment;

FIG. **10** shows the affect of the clipping region when used to clip the image in accordance with a preferred embodiment;

FIG. **11** represents an existing image upon which the clipped image will be applied in accordance with a preferred embodiment;

FIG. **12** indicates the result of applying the non-clipped image to the previous image;

FIG. **13** indicates which portions of the previously existing image are affected by applying the clipping region in accordance with a preferred embodiment;

FIG. **14** shows the composite result of applying the pixel image along with the composited underlying image, and the clipping region in accordance with a preferred embodiment;

FIG. **15** is a schematic block diagram that represents the operation of the invention in accordance with a preferred embodiment;

FIG. **16** is a schematic block diagram that represents the flow of graphical objects into a printing device in accordance with a preferred embodiment;

FIG. **17** is a flowchart illustrating the overall operation of a preferred embodiment;

FIG. **18** is a flowchart describing the method of processing printer non-imageable graphical objects in accordance with a preferred embodiment;

FIG. **19** is a flowchart illustrating the test determining if the non-imageable object is to be transformed into a collection of imageable objects or into a pixel object in accordance with a preferred embodiment; and

FIG. **20** is a flowchart illustrating the steps to create the pixel object in accordance with a preferred embodiment.

DETAILED DESCRIPTION OF THE ILLUSTRATIVE EMBODIMENTS

The invention is preferably practiced in the context of an operating system resident on a personal computer such as the IBM PS/2, or Apple, Macintosh, computer. A representative hardware environment is depicted in FIG. **3**, which illustrates a typical hardware configuration of a computer **300** in accordance with the subject invention. The computer **300** is controlled by a central processing unit **302**, which may be a conventional microprocessor; a number of other units, all interconnected via a system bus **308**, are provided to accomplish specific tasks. Although a particular computer may only have some of the units illustrated in FIG. **3** or may have additional components not shown, most computers will include at least the units shown.

Specifically, computer **300** shown in FIG. **3** includes a random access memory (RAM) **306** for temporary storage of information, a read only memory (ROM) **304** for permanent storage of the computer's configuration and basic operating commands and an input/output (I/O) adapter **310** for connecting peripheral devices such as a disk unit **313** and printer **314** to the bus **308**, via cables **315** and **312**, respectively. The function provided by cables **315** and **312** may be implemented via a network connection. A user interface adapter **316** is also provided for connecting input devices, such as a keyboard **320**, and other known interface devices including mice, speakers and microphones to the bus **308**. Visual output is provided by a display adapter **318** which connects the bus **308** to a display device **322** such as a video monitor. The workstation has resident thereon and is controlled and coordinated by operating system software such as the Apple System/7, operating system.

In a preferred embodiment, the invention is implemented in the C++ programming language using object-oriented programming techniques. C++ is a compiled language, that is, programs are written in a human-readable script and this script is then provided to another program called a compiler which generates a machine-readable numeric code that can be loaded into, and directly executed by, a computer. As described below, the C++ language has certain characteristics which allow a software developer to easily use programs written by others while still providing a great deal of control over the reuse of programs to prevent their destruction or improper use. The C++ language is well-known and many

articles and texts are available which describe the language in detail. In addition, C++ compilers are commercially available from several vendors including Borland International, Inc. and Microsoft Corporation. Accordingly, for reasons of clarity, the details of the C++ language and the operation of the C++ compiler will not be discussed further in detail herein.

As will be understood by those skilled in the art, Object-Oriented Programming (OOP) techniques involve the definition, creation, use and destruction of "objects". These objects are software entities comprising data elements and routines, or functions, which manipulate the data elements. The data and related functions are treated by the software as an entity and can be created, used and deleted as if they were a single item. Together, the data and functions enable objects to model virtually any real-world entity in terms of its characteristics, which can be represented by the data elements, and its behavior, which can be represented by its data manipulation functions. In this way, objects can model concrete things like people and computers, and they can also model abstract concepts like numbers or geometrical designs.

As will be recognized by those skilled in the art, the instant invention can be implemented in a procedural programming environment as well as an object-oriented one.

Objects are defined by creating "classes" which are not objects themselves, but which act as templates that instruct the compiler how to construct the actual object. A class may, for example, specify the number and type of data variables and the steps involved in the functions which manipulate the data. An object is actually created in the program by means of a special function called a constructor which uses the corresponding class definition and additional information, such as arguments provided during object creation, to construct the object. Likewise objects are destroyed by a special function called a destructor. Objects may be used by using their data and invoking their functions.

The principle benefits of object-oriented programming techniques arise out of three basic principles; encapsulation, polymorphism and inheritance. More specifically, objects can be designed to hide, or encapsulate, all, or a portion of, the internal data structure and the internal functions. More particularly, during program design, a program developer can define objects in which all or some of the data variables and all or some of the related functions are considered "private" or for use only by the object itself. Other data or functions can be declared "public" or available for use by other programs. Access to the private variables by other programs can be controlled by defining public functions for an object which access the object's private data. The public functions form a controlled and consistent interface between the private data and the "outside" world. Any attempt to write program code which directly accesses the private variables causes the compiler to generate an error during program compilation which error stops the compilation process and prevents the program from being run.

Polymorphism is a concept which allows objects and functions which have the same overall format, but which work with different data, to function differently in order to produce consistent results. For example, an addition function may be defined as variable A plus variable B (A+B) and this same format can be used whether the A and B are numbers, characters or dollars and cents. However, the actual program code which performs the addition may differ widely depending on the type of variables that comprise A and B. Polymorphism allows three separate function definitions to be written, one for each type of variable (numbers, characters and dollars). After the functions have been defined, a program can later refer to the addition function by its common format (A+B) and, during compilation, the C++ compiler will determine which of the three functions is actually being used by examining the variable types. The compiler will then substitute the proper function code. Polymorphism allows similar functions which produce analogous results to be "grouped" in the program source code to produce a more logical and clear program flow.

The third principle which underlies object-oriented programming is inheritance, which allows program developers to easily reuse pre-existing programs and to avoid creating software from scratch. The principle of inheritance allows a software developer to declare classes (and the objects which are later created from them) as related. Specifically, classes may be designated as subclasses of other base classes. A subclass "inherits" and has access to all of the public functions of its base classes just as if these function appeared in the subclass. Alternatively, a subclass can override some or all of its inherited functions or may modify some or all of its inherited functions merely by defining a new function with the same form (overriding or modification does not alter the function in the base class, but merely modifies the use of the function in the subclass). The creation of a new subclass which has some of the functionality (with selective modification) of another class allows software developers to easily customize existing code to meet their particular needs.

Although object-oriented programming offers significant improvements over other programming concepts, program development still requires significant outlays of time and effort, especially if no pre-existing software programs are available for modification. Consequently, a prior art approach has been to provide a program developer with a set of pre-defined, interconnected classes which create a set of objects and additional miscellaneous routines that are all directed to performing commonly-encountered tasks in a particular environment. Such pre-defined classes and libraries are typically called "application frameworks" and essentially provide a pre-fabricated structure for a working application.

For example, an application framework for a user interface might provide a set of pre-defined graphic interface objects which create windows, scroll bars, menus, etc. and provide the support and "default" behavior for these graphic interface objects. Since application frameworks are based on object-oriented techniques, the pre-defined classes can be used as base classes and the built-in default behavior can be inherited by developer-defined subclasses and either modified or overridden to allow developers to extend the framework and create customized solutions in a particular area of expertise. This object-oriented approach provides a major advantage over traditional programming since the programmer is not changing the original program, but rather extending the capabilities of the original program. In addition, developers are not blindly working through layers of code because the framework provides architectural guidance and modeling and, at the same time, frees the developers to supply specific actions unique to the problem domain.

There are many kinds of application frameworks available, depending on the level of the system involved and the kind of problem to be solved. The types of frameworks range from high-level application frameworks that assist in developing a user interface, to lower-level frameworks that provide basic system software services such as communications, printing, file systems support, graphics, etc. Commercial examples of application frameworks

include MacApp (Apple), Bedrock (Symantec), OWL (Borland), NeXT Step App Kit (NeXT), and Smalltalk-80 MVC (ParcPlace).

While the application framework approach utilizes all the principles of encapsulation, polymorphism, and inheritance in the object layer, and is a substantial improvement over other programming techniques, there are difficulties which arise. These difficulties are caused by the fact that it is easy for developers to reuse their own objects, but it is difficult for the developers to use objects generated by other programs. Further, application frameworks generally consist of one or more object "layers" on top of a monolithic operating system and even with the flexibility of the object layer, it is still often necessary to directly interact with the underlying operating system by means of awkward procedural calls.

In the same way that an application framework provides the developer with prefab functionality for an application program, a system framework, such as that included in a preferred embodiment, can provide a prefab functionality for system level services which developers can modify or override to create customized solutions, thereby avoiding the awkward procedural calls necessary with the prior art application frameworks programs. For example, consider a printing framework which could provide the foundation for automated pagination, pre-print processing and page composition of printable information generated by an application program. An application software developer who needed these capabilities would ordinarily have to write specific routines to provide them. To do this with a framework, the developer only needs to supply the characteristics and behavior of the finished output, while the framework provides the actual routines which perform the tasks.

A preferred embodiment takes the concept of frameworks and applies it throughout the entire system, including the application and the operating system. For the commercial or corporate developer, systems integrator, or OEM, this means all of the advantages that have been illustrated for a framework such as MacApp can be leveraged not only at the application level for such things as text and user interfaces, but also at the system level, for services such as printing, graphics, multi-media, file systems, I/O, testing, etc.

FIG. **4** shows a schematic overview of a computer system utilizing the object-oriented printing interface that uses the present invention. The computer system is shown generally as box **400**, and an application program **402** and an operating system **406** are provided to control and coordinate the operations of the computer. Application program **402** communicates with operating system **406** as indicated by arrow **404**. However, in accordance with an embodiment of the invention, rather than communicating directly with a standard interface, such as grafport **410**, application program **402** can now communicate with operating system **406** at a higher level when it wishes to print information. This latter interaction is accomplished by providing an object-oriented printing interface shown schematically as box **424**. Thus, the application program **402** sends graphic objects to the object-oriented printing interface **424**. These graphic objects include those that define both textual and graphical images. Printing interface **424** responds to these objects in order to perform various formatting and pagination functions. The formatted information is then transmitted to a grafport **410** as indicated schematically by arrow **426**.

In any case, the information flows through grafport **410**, and as indicated by arrow **412**, is provided to a printer handler **414**. Printer handler **414** is similar to printer drivers previously described. However, it is "intelligent" and offers some additional capabilities which will be described herein. Essentially, printer handler **414** processes the incoming data stream of graphic objects indicated by arrow **412** and adds the necessary printer commands to control the associated printer schematically illustrated as printer **422**. The resulting printer data stream is provided, as indicated by arrow **416**, to a printer port **418** which converts the data into electrical signals that are sent over data pathway **420** to printer **422** which processes the data and generates an image responsive to the data on the print medium.

The actual mechanism by which a document generated by the printing interface **424** is transmitted to printer **422** is shown in more detail in FIG. **5**. In particular, printing interface **424** generates an entity called a document folio shown schematically as document folio **500** in FIG. **5**. The document folio consists of graphics objects that include objects that define textual images, objects that define graphical images or a combination of the two, all formatted and arranged in a manner specified by the application program. The document folio information is provided, as indicated by arrow **512**, to the spooling system **501** by presentment to the print channel **502**. Print channel **502** is an object which is created to transport the information to an appropriate printer. Print channel uses a print job description and a printer identification provided by the application program to transmit the printable information to the appropriate printer.

More specifically, after the print channel **502** receives a printing job, it transmits the printable information to a spooler program **518** as indicated by arrow **514**. Spooler **518** receives the incoming information stream and stores it, in incoming order, in an intermediate storage location **522** as indicated by arrow **520**. Print channel **502** then sends a notification to a print server program **506** via a link **504**, which notification informs print server program **506** that a new print job exists. The print server program **506** is a standard program which monitors all print jobs that have been created and also checks the available printers to determine their status.

Once a print job has been completely spooled or stored in intermediate storage **522**, the print server **506** notifies a printer handler **510** by means of a link **508**. The printer handler **510** is a type of printer driver which controls and drives a specific printer; its purpose is to convert textual and graphical objects into printer readable form for any particular printer type—to create a printer data stream, indicated schematically by arrow **530**. Typically, a printer handler can only process one print job and any other jobs that are created and directed towards the associated printer are queued for later processing. The printer handler contains a de-spooler program (indicated as box **516**) which retrieves the spooled data from intermediate storage **522** (as indicated by arrow **524**) and provides the information, as indicated by arrow **526**, to an imaging engine schematically illustrated as box **532**. The imaging engine **532** converts the incoming data stream into the command signals which are necessary to drive the printing elements to produce the final printed document. The string of commands make up the printer data stream **530**. The printer data stream **530** is provided to the actual print device indicated by box **528** for printing.

These command signals are printer dependent. Many different printers support one or more common page description languages (PDLs) such as PostScript or PCL. The printer command primitives provided by the PDL must include a primitive that provides the printer with the ability to receive a pixelmap to be printed. Example PostScript PDL commands that would provide this facility are the "image" and "imagemask" commands.

A pixelmap is a rectangular array of pixels as shown in FIG. **8** by the box **801** surrounding the pixelated image **802**. Each pixel, one of which is indicated by **803**, potentially has a characteristic such as color or intensity but which may be as simple as a binary value. For this example, the group of three pixels at **804** are intended to be transparent.

Some graphical objects may have the characteristic that they cannot be directly imaged by the PDL supported by the print device **528**. The instant invention is a means to detect these objects—non-imageable objects—and to transform them into a group of imageable objects so that the desired image corresponding to the object is printed. This is the subject matter of the instant invention and is further described below.

Once a print job is completely printed, the printer handler **510** checks its queue for another print job and, if a job is present, begins processing it. Alternatively, if there are no new print jobs to process, the printer handler becomes inactive. After a particular print job is completed, the information stored in intermediate storage in **522** is deleted and the storage is reclaimed. As an option, the storage can be retained until a user explicitly deletes the storage.

The printer handler framework facilitates creation of frameworks for different types of printers like PostScript printers, raster, vector, and PCL printers. A preferred embodiment provides a framework that is extensible so that new printers can be added to the system very easily. This includes printers with new imaging models, communication protocols, etc. The preferred embodiment also provides a framework that does most of the work to support a printer and at the same time provides enough flexibility for developer customization. The preferred embodiment also allows developer customization at various times during the printing process. Customization occurs at the desktop printer level for presenting device specific user interface, at print time for setting print time features (like duplex printing, multi-bin paper feeding, etc.), at imaging time, by providing a way to access the device at the beginning/end of document and page, and by providing a way to customize rendering of graphics model primitives. Finally, at the device level to support different communication protocol(s).

Clients

All printer manufacturers are clients of the printer handler framework. A client uses a framework to design PostScript, raster, vector, and PCL printer handler frameworks. Developers start from one of these special types of frameworks and customize it for their printers. For example, a PostScript printer developer (like QMS or Linotype) would use the PostScript printer framework, a plotter developer (like CalComp) would customize the vector printer handler framework.

Architecture

The desktop printer is the only user visible part of the printer. All user actions (commands) related to the printer are handled by this object. When a document is to be printed, it is dragged to the printer icon (or the Print item is selected from the system wide menu). This starts the printing process by first spooling the document by packaging it as a print job. Spooling is a process of streaming the graphic description of a document to the disk so it can be imaged later (possibly in a different task) to the actual printing device represented by the desktop printer. The spooled print job is stored in the printer's persistent data which consists mainly of the print job queue and the printer's dynamic configuration. After the print job is spooled, the printer handler is sent a message that there is a print job for it to process. The printer handler picks up the print job, despools it, converts it to the printer's native imaging model and sends it to the printer.

User's Model of Printing

A reference to a printer in a preferred embodiment really mean a printer model and its associated files, which includes the printer handler. A model is a class that is subclassed from an existing class. Since all user visible entities on a system are models, or the interface presented by them, it makes sense to talk about a printer this way. Each printer model keeps its persistent data (job queue, configuration, etc.) in its model store. The model store provides a way for a model to create separate files that it alone manages. It has protocol for interfacing to a file system. The printer "component" has certain dependencies that must be satisfied when it is moved between machines or enclosed in a business card. A printer is typically dependent on its personality document, the shared library and archive for the system classes that implement the printer, and the shared library and archive for the developer supplied customizations.

When a user "installs" a printer handler in the system, it is immediately available for direct connect devices or in the network browser for network devices. This processing is facilitated by creating a physical model for a direct connect device and a service adapter is "registered" for a network device. A physical device object represents a real device that can be connected directly to the computer (as opposed to available on the network). It is capable of creating a subclass that represents the device. A service adapter indicates the directory protocols (AppleTalk Name Binding Protocol (NBP), Domain Naming System (DNS) etc.) and service identifiers (LaserWriter) it can handle and is called on by a directory service to create a model for a service available on a physical directory. To print to a direct connect device, a user connects the printer to the computer (for serial devices) and then drags a document to it. To print to a network device, either the document is dragged to the printer in the network browser or the printer is dragged to the desktop and then the document is dragged to it.

Printer Personality Document

A personality document is supplied by the device manufacturer. The printer personality "document" contains instances of objects, just the data, that implement a particular printer. In other words, it is a shared library without code—just the archive. Examples of objects in the personality document are the printer configuration, printer resolution, print job description which specifies the print time options available on the printer, and the print device object that converts the graphic data into the printer's imaging model. FIG. **6** shows how the personality document is used by different parts of the printing system **600**. The personality document **601** supplied by the developer is used in read-only mode by the printing system **600**. The desktop printer **605** and the printer handler **603** "read" this document to access its personality objects polymorphically. The desktop printer **607** accesses the printer personality document **601** as shown by arrow **612** to obtain printer dialogs. The desktop printer **607** communicates with the physical printer **605** indicted by arrow **608** to obtain the physical printer's configuration, setup, and other similar printer information. The desktop printer **605** maintains the dynamic state of the printer, the print job queues, status, and other relevant information in the printer's persistent state **604** as indicated by arrow **609**. This information in turn is accessed by the printer handler **603** as shown by arrow **610**. Printer handler **603** converts the queued data into a form suitable for the physical printer **605** and sends the converted data to the physical printer **605** as schematically indicated by arrow **611**.

A printer personality document provides the printer's attributes. One of these attributes is a complexity threshold.

This threshold may be determined from other attributes of the printer such as the printer model, memory capability and page description language (PDL). The graphic primitives that can be used with the printer are determined from the page description language used with the printer. To implement the instant invention, the PDL must support a pixelmap primitive. This primitive provides the facility for transmitting a pixel by pixel representation of a portion of the image to be printed to the printer at the resolution of the printer. This primitive allows a pixelmap, as illustrated in FIG. **8** at **801** to be inserted into the printer data stream **611**.

The analogy of a printer model and its personality to an application and its document implies that a printer model can "read" many personality documents. However, in most cases there is only one personality document **601** per printer because a printer model represents one physical printer **605**. In the case where the user has more than one printer of the same type (for example, two LaserWriter IIg printers), one personality document may be "shared" by multiple printers. The desktop printer **607** obtains user interface objects from the personality (the objects that present the user with printer configuration, features and settings that can be manipulated). The printer handler **603** gets imaging objects from the personality and calls on them to reproduce the document on the printer **605**. Once the printer's dynamic state is added to its persistent store **604**, both the desktop printer **607** and the printer handler **603** refer to it for the printer's configuration. A personality document **601** is created for each type of printer that a printer handler **603** is created for. The document is created and given to the developer of that type of printer. The developer can "edit" the instance data of objects or replace them with the printer specific implementations.

Printer Handler

FIG. **7** details the printer handler components and their interactions in accordance with a preferred embodiment. The printer handler server team is started by the desktop printer when a print command is initiated by either dragging the document to it or selecting the Print command from one of the menus. The printer handler program creates a dispatch giving it a dispatcher and a message stream to be used as a transport between the client and the server. The dispatch task combines the transport and the dispatcher to create the printer handler server.

There is one printer handler task per physical printer. The printer handler consists of a dispatcher and an imaging task. The dispatcher is a task that handles requests to print jobs and sends them to the imaging task so that the server task is free to handle other requests (more print jobs, status queries, notification, etc.). The printer handler architecture allows for more than one task working on print jobs. Instead of having only one imaging task, the printer handler dispatcher can have a pool of tasks that access the job queue and pick up a job to process. This way, the printer handler can be processing multiple jobs at the same time. If the target printer can accept only one job at a time, only one imaging task will be talking to it at a given time. The multiple imaging tasks model works well when the destination is a spooler that can accept more than one job at a time. Each job in the queue will know the task that is processing it so things like abort, pause, etc. can function properly.

Printer Handler Server

The printer handler task is started by the desktop printer when a print command is initiated by either dragging the document to the printer icon or selecting the Print command from one of the menus. The desktop printer has a client class that starts up the server. The client class provides the protocol for calling "remote" procedures that are implemented in other objects, tasks, or servers. The printer handler program creates a transport and a dispatcher to create the printer handler server.

There is one printer handler task per physical printer. The printer handler consists of a dispatcher and an imaging task. The dispatcher handles requests to print jobs and sends them to the imaging task so that the server task is free to handle other requests (more print jobs, status queries, notification, etc.). The instant invention is within the imaging task. The printer handler architecture allows for more than one task working on print jobs. Instead of having one imaging task, the printer handler dispatcher has a pool of tasks that access the job queue and pick a job to process. This way, the printer handler can be processing multiple jobs at the same time. If the target printer can accept only one job at a time, only one imaging task communicates to it at a given time. The multiple imaging tasks function efficiently when the destination is a spooler that can accept more than one job at a time. Each job in the queue understands the task that is processing it so things like abort, pause, etc. are managed properly.

Printer Handler Imaging Task

The printer handler imaging task receives one print job at a time. It uses the developer customizable imaging objects to convert the source description of the document into a stream of data targeted for a particular printer. The portion of this task that is relevant to the invention is illustrated in FIG. **15**. The imaging task obtains a spool file, shown schematically by arrow **1501**, from the print job queue—the function of the de-spooler **1502**—and passes it to the printer device object, TPrintDevice **1503**. The printer device object **1503** reads the spool file and extracts individual pages from it and directs the conversion of the individual pages into the printer's imaging model using the printer's PDL. The imaging objects also perform the task of mapping the attributes requested by the print job (page size, media, color, fonts, etc.) to features actually available on the printer. This processing is achieved by consulting the printer's dynamic state maintained by the printer handler.

The imaging objects produce an output stream—the printer data stream **1516**—that is sent to the output device. The framework for a specific type of printer defines an appropriate class. For example, the PostScript printer handler framework defines a class whose subclass talks with a printer using the AppleTalk Printer Access Protocol (PAP). A TCP/IP based printer can be easily supported by subclassing the same class.

Architecture of a Preferred Embodiment

A preferred embodiment addresses the problem of converting extremely functional graphical objects into less functional display device primitives. An example printer device is illustrated in FIG. **16**. Graphical objects **1602** are sent to a spooling system **1603** as indicated by arrow **1602**. The spooling system **1603** converts the graphical objects **1602** into printer command primitives **1605** that are sent to printer **1606** as shown by arrow **1604**. Graphical objects **1602** consists of two types of objects, non-imageable objects **1607** and imageable objects **1608**. Imageable objects **1608** are directly transformed into printer command primitives **1605**. Non-imageable objects **1607** cannot be directly transformed into printer command primitives **1605** and require significant processing to transform these graphical objects **1607** into printer command primitives **1605**.

A preferred embodiment utilizes a process where the printer device object **1503**, described above and illustrated in FIG. **15**, determines which graphical images are non-imageable by the printer device, determines the complexity

of the non-imageable graphical object and analyzes the best method to convert the non-imageable object to imageable object(s). The following description is directed toward a PostScript capable printer, but as would be recognized by one skilled in the art, the description and the invention are not limited to PostScript capable printers, and supports display devices.

PrintServer **506** in FIG. **5** sends a notification to TPrinterHandler **1504** that a print job exists in its queue. TPrinterHandler **1504** opens the spooled file on the intermediate storage **522** device using de-spooler **1502** and passes the spooled file to TPrintDevice **1503**. TPrintDevice **1503** reads the spool file, and sends the graphical objects—a page at a time—to TPostScriptGrafDevice **1506** (as shown schematically by arrow **1505**). TPostScriptGrafDevice **1506** ignores non-imageable objects, but converts the imageable objects in the page into PostScript PDL commands that make up the printer data stream **1516** that is sent to print device **1517**. The page is then passed to the TPostProcessingGrafDevice object **1514**. TPostProcessingGrafDevice **1514** at box **1508** rereads the page's graphical objects searching for non-imageable graphic objects. If the page consists only of imageable objects, the page definition for the printer data stream is closed as shown schematically by arrow **1518**. If the page contains a non-imageable graphical object, the graphical object and its bounds are saved and its complexity initialized.

The complexity is further adjusted by detecting which imageable objects, previously defined in the page, intersect with the non-imageable object. TPostProcessingGrafDevice **1514** at box **1508** rereads the page graphical objects once again to detect which of the imageable objects intersect with the bounds of the saved non-imageable graphical object. The complexity of the non-imageable object is increased for each intersection depending on the complexity of the object intersecting the bounds of the non-imageable object. Thus, developing a measure of complexity and an association of what imageable objects interact with the non-imageable object for that page. This continues for all non-imageable graphical objects defined in the page.

Once all of the page's non-imageable graphical objects and the imageable objects that intersect with the non-imageable objects have been analyzed, the complexity of each saved grouping is compared to the complexity threshold of the printer. If the complexity is more than the threshold, the saved grouping is sent to the NativeImager object **1510** as shown by arrow **1509**. Otherwise, the saved grouping is sent to the OutlineMakerVertexEngine object **1513** as shown by arrow **1512**. The complexity is determined by the number of line segments required to draw the image. Because the printer has limits on the number of line segments per page (this limit is one of the components used to determine the printer's threshold) attempting to draw more line segments than that limit will cause errors in the resultant printed image. Thus the invention monitors the number of line segments per page (the complexity) and compares this with the capabilities of the printer (the threshold).

The NativeImager object **1510** constructs a rectangular pixelmap of the image, in RAM **306**, defining each pixel the image to be printed. This pixelmap is created at the resolution of the printer as determined from the Printer Personality Document and/or the Printer's Persistent Data. The NativeImager object **1510** also defines a clipping region that outlines the image within the rectangular pixelmap. This clipping outline is used to effectuate only the image defined within the rectangular pixelmap instead of the entire rectangular pixelmap so that the pixelmap can be applied without affecting the prior images that overlap the rectangular area of the pixelmap but that do not intersect with the image contained within the pixelmap. This rectangular pixelmap and a clipping outline are appended to the printer data stream **1516** for the page by sending the rectangular pixelmap and clipping outline to TPostScriptGrapfDevice **1506** as shown schematically by arrow **1511**.

FIG. **8** shows a pixelmap **801** containing a pixelated complex image **802**. This image is for illustration only, it is not intended to indicate a high complexity image. The pixel indicated by arrow **803**, although part of the pixelmap, is not needed to render the pixelated image **802**. In fact, if the pixelmap **801** is actually applied over an existing image in a manner that applies pixel **803**, pixel **803** could obscure the underlying image. The affect of this is shown in FIGS. **11** and **12**. In FIG. **11**, box **1106** indicates the partially imaged page that already contains lines **1107** and **1108**. Box **1101** indicates where the pixelmap will be placed and showing that the pixelmap will overlap parts of lines **1107** and **1108**. FIG. **12** shows the result of placing the pixelmap **1201** over the partially imaged page **1206** showing that lines **1207** and **1208** are obscured by pixelmap **1201**.

FIG. **9** shows a clipping region **905** surrounding the pixelated complex image **902**. FIG. **10** shows just the clipping region **1005**. The purpose of developing clipping region **905** is to allow the rectangular pixelmap **1201** to be applied to the page without affecting any area of the page other than the area of the complex pixelated image **1202**.

As mentioned above, FIG. **12** shows image **1206** that would be partially covered by rectangular pixelmap **1201** if the pixelmap were simply applied over image **1206**. With the clipping region **1205** applied to the pixelmap **1201** only the portion of the pixelmap **1301** that actually contains the image is composited into the image **1206**. For example, line segment **1207** would have been overlaid by the application of the pixelmap if applied without the clipping region. Thus obliterating portions of line segment **1207**. Because of the gap **1204** is transparent, portions of line **1208** must be included in the pixelated image **1202**.

FIG. **13** demonstrates the result obtained by clipping pixelmap **1301**. Now line **1307** is not affected by the application of pixelmap **1301**. However, line **1308** is affected as the line segment **1309** is still obliterated by clipping region **1305**. Thus in the generation of the pixelated image such as **802** shown in FIG. **8**, previously imaged objects that intersect with image **802** must also be imaged into the image **802**. Thus resulting in the image shown in FIG. **14**.

FIG. **14** shows the final composited image **1406**. Now because of the insertion of line segment **1409** into the pixelated image **1402**, the image is complete. The dotted box **1401** indicates the placement of the pixelmap, but the dotted lines are not part of the composited image. Thus, the non-imageable object has been transformed into a collection of imageable objects corresponding to the original object **1519**.

The OutlineMakerVertexEngine object **1513** transforms the original graphical object into a collection of imageable objects corresponding to the original object **1519** that can be processed by TPostScriptGrapfDevice **1506** as compared to the original graphical object that cannot be processed by TPostScriptGrapfDevice **1506**. This collection of objects is sent to TPostScriptGrapfDevice **1506** as indicated by arrow **1515** where the imageable objects are converted into PostScript and appended to the printer data stream **1516** for the page.

FIG. 17 is a flow chart representation of the embodiment described above. Terminal 1701 is entered when the spooling system notifies the printer handler imaging task that file is ready to print. In step 1702, the file is opened and passed to step 1703. Step 1702 continues to administrate additional notifications from the spooling system of any additional files to be printed. After the current file is completely processed, step 1702 opens the next queued file and repeats the operation described below. If no additional files are to be printed the printer handler imaging task becomes quiescent.

Step 1703 reads the file and indicates page boundaries within the file. The information for each page is passed to step 1704. Step 1704 sends the page date to the PostScript-GrafDevice object where the page date is converted to the PostScript page description language (PDL). The same page information is sent to step 1705 where any non-imageable graphical objects are transformed into imageable graphical objects and, as will be shown below, transformed into the PostScript PDL. Finally, step 1706 closes the PDL page, and the next page is processed starting at step 1703. This process repeats for all pages in the file. When the file is completely processed, step 1703 communicates with step 1702 to begin processing the next spooled file if any.

The details of step 1705 are shown by the flowchart shown in FIG. 18. Processing starts at terminal 1801. Step 1808 examines each graphical object in the page. Decision 1804 determines whether the object is an imageable or non-imageable object. If the object is imageable, the next object is examined. Alternatively, processing continues at step 1806 where the non-imageable object is stored, its bounds determined by step 1807, and the complexity of the non-imageable object initialized by step 1802. Next step 1809 examines all previous objects in the page to determine which objects intersect with the bounds of the non-imageable object saved in step 1806. This determination is made at decision 1811. If an object is found that intersects the bounds of the non-imageable object, the non-imageable objects complexity is updated at step 1812 and the intersecting object is associated with the non-imageable object in step 1810. This process repeats until all the objects in the page prior to the non-imageable object are examined, the complexity of the non-imageable object determined, and all intersections with prior objects with the non-imageable object associated with the non-imageable object. Step 1805 transforms the non-imageable object into the PDL as will be described below. After the non-imageable object is transformed, the process continues at step 1808 until the entire page is examined.

FIG. 19 contains a flowchart detailing the operation of step 1805. Entry is through terminal 1901. Decision 1902 compares the complexity of the non-imageable object to the complexity threshold of the device associated with the printer handling imaging task. If the complexity is below the threshold, processing continues at step 1904 where the non-imageable object is decomposed in to a collection of imageable objects. These imageable objects converted to the PostScript PDL by PostScriptGrafDevice as indicated by step 1905. Alternatively, if the complexity of the non-imageable object is not below the threshold, processing moves to step 1903 where a pixelated image is created from the non-imageable object and its associated intersecting objects. The pixelated image and clipping region objects are also converted to the PostScript PDL at step 1905. Finally, processing returns through terminal 1906.

Processing for step 1903 is shown by the flowchart on FIG. 20. Entry is via terminal 2001. Step 2002 creates a pixel map of the printer non-imageable object. Step 2003 adds the intersecting portions of the intersecting objects to the pixel map. The pixel map is converted to a pixel object in step 2004. Step 2005 creates a clipping object that outlines the pixelated image of the non-imageable object and the intersecting portions of the associated objects. This clipping region is finally converted to a clipping object in step 2006. Processing returns through terminal 2007.

Printer's Persistent Data

The printer handler is responsible for keeping track of the printer's persistent data, in particular, its job queue and dynamic state. The dynamic state contains things like the current media/tray setting, current color mode (2, 3 or 4 colors), printer resolutions (for variable resolution printers), etc. Since each printer would want to save different things in its dynamic state, there is a class that developers can subclass to add printer specific features. For the most part, it is this class that will be streamed out to the printer's persistent data store. The default implementation of the persistent data store will be a dictionary, although the developer is free to use some other storage scheme. Since the printer's state is persistent, the printer handler can be easily restarted in case of a crash or system shutdown.

The printer state, which is part of the printer model's store, is updated when the printer handler images a job to it and finds that the state has changed. This scheme works when the printer handler has two-way communication available with the printer. In cases when two-way communication is not available, the printer handler will rely on the printer's static configuration.

Printer Personality Document Excerpts

The following are selected entries from an Adobe PostScript Printer Description File which illustrates the features of the PPD that can be used to determine a PostScript printer's attributes.

```
*FormatVersion: "4.0"
*FileVersion: "2.8"
*PCFileName: "A_PNT518.PPD"
*LanguageVersion: English
*Product: "(LaserWriter Personal NT)"
*PSVersion: "(51.8) 0"
*ModelName: "Apple LaserWriter Personal NT"
*NickName: "LaserWriter Personal NT v51.8"
```

The above excerpt provides basic information regarding the format of the PPD and specific information about the associated printer. Specifically, the product name and the version of PostScript supported by the printer.

```
*% General Information and Defaults ==========
*FreeVM: "402366"
*LanguageLevel: "1"
*ColorDevice: False
*DefaultColorSpace: Gray
*VariablePaperSize: False
*FileSystem: False
*Throughput: "4"
. . .
*DefaultResolution: 300dpi
. . .
```

The above provides specific information regarding the characteristics of the printer. These include the "FreeFM" which limits the complexity of a PostScript job. The "LanguageLevel" value determines what printer command primitives are supported by the printer and how many line segments are allowed in a path, and the total number of line segments allowed on a page. The above also indicates the printer's default resolution. This value is used to determine

the size of a pixelmap that will hold a non-imageable object. Additional information includes whether the printer is a color device, whether the printer includes its own file system, and the printer's maximum throughput in pages per minute.

Printer Handler Classes

A discussion of the classes that implement the printer handler framework is provided below.

TPrinterHandler

```
class TPrinterHandler : public MRemoteDispatcher {
    public:
        TPrinterHandler ( );
        virtual~TPrinterHandler( );
    private:
    // Server Requests
    // Every XXXRequest method unflattens arguments, if any, and then
    // calls the corresponding HandleXXX method. It then calls
    // ReturnSuccess and flattens results to the result stream.
        // Job Queue Management
        // Requests that apply to all jobs in the queue
            void      GetAlljobsRequest ( );
            void      AbortAllJobsRequest( );
            void      DeferAllJobsRequest( );
            void      UndegerAllJobsRequest( );
            void      DeferAllJobsUntilRequest( );
        // Requests that apply to one job in the queue
            void      AddJobRequest( );
        // RemoveJobRequest will abort the job if it is currently being processed.
        // Otherwise, the job will be removed from the queue.
            void      RemoveJobRequest( );
            void      PauseJobRequest( );
            void      DeferJobRequest( );
            void      UndeferJobRequest( );
        // Update printer's state
            void    UpdateDynamicPrinterDataRequest( );
        // return status of a job
            void      GetStatusOfJobRequest( );
        //
        // Subclasses can override the following HandleXXX methods.
        // HandleXXX are called from the corresponding request methods.
        //
        // Job Queue Management
        // Requests that apply to all jobs in the queue
        virtual void    HandleGetAllJobs( );
        virtual void    HandleAbortAllJobs( );
        virtual void    HandleDeferAllJobs( );
        virtual void    HandleUndeferAllJobs( );
        virtual void    HandleDeferAllJobsUntil( );
        // Requests that apply to one job in the queue
        virtual void    HandleAddJob( );
        // RemoveJobRequest will abort the job it it is currently being processed.
        // Otherwise, the job will be removed from the queue.
        virtual void    HandleRemoveJob( );
        virtual void    HandlePauseJob( );
        virtual void    HandleDeferJob( );
        virtual void    HandleUndeferJob( );
        // Update printer's state
        virtual void    HandleUpdateDynamicPrinterData( );
        // return status of a Job
        virtual void    HandleGetStatusOfJob( );
        // for TPrinterHandlerCaller only
        typedef enum {
            kGetAllJobsRequest, kAbortAllJobsRequest, kDeferAllJobsRequest,
            kUndeferAllJobsRequest, kDeferAllJobsUntilRequest,
            kAddJobRequest, kRemoveJobRequest, kPauseJobRequest,
            kDeferJobRequest, kUndeferJobRequest,
            kUpdateDynamicPrinterDataRequest,
            kGetStatueRequest
        };
        friend class TPrinterHandlerCaller; // so it can use enums above.
    protected:
        // Get the imaging task to send jobs to
        virtual TPrinterImagingTask*   GetImagingTask( );
        // Get the job queue for this printer
        virtual TDeque* GetPrintJobQueue( );
        // . . . . . Methods to communicate with the imaging task
        // . . . . . .
```

TPrinterHandler is a base class that provides protocol for dispatching server requests. The corresponding client class TPrinterHandlerCaller is described later. TPrinterHandler maintains a job queue for the printer. This queue is semaphore protected to allow concurrent access by the printer handler and an imaging task.

GetImagingTask creates a TPrinterImagingTask giving it an exchange to communicate with (an exchange provides a place to send messages to and receive messages from). TPrinterImagingTask is given one job at a time to process (by AddJobRequest). When the job is finished, it notifies the printer handler so it can decide what to do with the job.

GetStatusOfJobRequest returns status of a job in the job queue. For the job being processed currently, the status reported is the "global" job status described earlier. There are two ways this could be implemented. One way is for the TPrintDevice subclass to post the status (perhaps a TText) periodically at a global location which the printer handler returns to the client in GetStatusRequest method. Another way is to implement a helper task to get status from the TPrintDevice subclass. For any other job (not currently being processed), the status that is returned might be the number of pages in the job (if that's appropriate), how far down the queue this job is, etc.

TPrinterHandlerCaller

```
class TPrinterHandlerCaller : protected MRemoteCaller {
    public:
        TPrinterHandlerCaller(TSenderTransport*);
        virtual~TPrinterHandlerCaller( );
        // Remote requests
        // These are called by TPrinterModel's command handlers
        virtual TPrintJobQueue*   CreateJobIterator( );
        virtual void    AbortAllJobs( );
        virtual void    DeferAllJobs( );
        virtual void    UndeferAllJobs( );
        virtual void    DeferAllJobsUntil( );
        virtual void    AddJob(TPrintJobSurrogate&);
        virtual void    RemoveJob(const TPrintJobSurrogate&);
        virtual void    PauseJob(const TPrintJobSurrogate&);
        virtual void    DeferJobRequest(const TPrintJobSurrogate&);
        virtual void    UndeferJobRequest(const TPrintJobSurrogate&);
        virtual void    GetStatus(TText&);
        //.....
    MRemoteCallerDeclarationsMacro(TPrinterHandlerCaller);
};
```

A TPrinterHandlerCaller is instantiated in the printing task (the task that initiates printing, probably a compound document) by the TPrinterModel. It uses a transport to send a request to the printer handler task. The transport can be local or remotely located. Thus, the printer handler to be on a remote machine. A reference to an already-registered service (like a network printer) required by the transport is known to the printer that the document is being printed on. When the printer handler is remote, TServiceReference is obtained from the network.

TPrinterHandlerCaller's methods are called by the printer model's commands which are called by the document framework in response to user actions.

TPrinterImagingTask

```
class TPrinterImagingTask : public TTaskProgram {
    public:
        TPrinterImagingTask (TIPCExchange*);
        virtual          ~TPrinterImagingTask( );
        // TTaskProgram override
        virtual void     Run( );
        // Support methods to handle different types of messages from the
        // Printer Handler.
        // The messages that this task will receive are:
        //      AbortJob
        //      PauseJob
        //      GetStatus
        //      ProcessJob
        //      etc.
        //.....
};
```

TPrinterImagingTask, which is created by the printer handler, performs the task of imaging print jobs and supplying progress information for the same. The constructor receives a TIPCExchange that the imaging task uses to communicate with the printer handler. As far as the communication between the printer handler server and the imaging task is concerned, there are two possibilities. One is to use the exchange to receive messages and dispatch them based on the message id that the printer handler attaches to the header. Another way to do Inter-Process Communication (IPC) is to use wait groups to handle the dispatching automatically when you provide message handlers for each type of message. The second method makes the implementation more structured (avoids a switch statement) but involves writing more code.)

ProcessJob gets a TPrintJobSurrogate as a parameter. Using the TPrintJobSurrogate, the imaging task gets to the print job. A print job has a reference to the printer's persistent data (a TDiskDictionary) that keeps the spooled image of the document, a reference to the printer that the job was targeted to, etc. The printer reference (lets call it TPrinterIdentity) is actually a reference to the TPrinterModel's data. Using this data the imaging task can get to the TPrintDevice subclass for the printer. The imaging task gets the spool stream and the job description (TPrintJobDescription) out of the job's persistent data and asks the print device to process it.

TPrintDevice

```
class TPrintDevice : public MCollectible {
    public:
        virtual ~TPrintDevice( );
// Don't override these: override the HandleXXX methods below.
    virtual void RenderPrintRun(const TPrintRun&,const TPrintJobDescription&,
                const TPrinterIdentity&);
    virtual void RenderPage(const TGrafRun& grafRun, const TPageDescription&,
//MCollectible support
        virtual TStream& operator<<=(TStream& fromwhere);
        virtual TStream&operator>>=(TStream& towhere) const;
```

-continued

TPrintDevice

```
        virtual Boolean IsEqual(const MCollectible *) const;
    protected
        TPrintDevice& operator=(const TPrintDevice&);
        // You can use these in the HandleXXX methods below to get current
        // page/Job information.
        virtual TPrintRun* GetPrintRun( );
        virtual const TPrintJobDescription* GetPrintJobDescription( );
        virtual TGrafRun* GetGrafRun( );
        virtual TPageDescription* GetPageDescription( );
        // The following methods are called as a result of RenderPrintRun( ).
        // Don't call these directly: call RenderPrintRun( ).
        // You may override these. If you override Begin/EndPrintRun, then
        // be sure to call these base class methods as the first thing
        // in your override implementations.
        virtual void    HandleBeginPrintRun( );
        virtual void    HandleRenderPrintRun( );
                        // Default implementation goes through the printrun in
                        // forward order and calls RenderPage for each page.
        virtual void    HandleEndPrintRun( );
        // The following methods are called as a result of RenderPage( ).
        // Don't call these directly: call RenderPage( ).
        virtual void    HandleBeginPage( );
        virtual void    HandleRenderPage( ) = 0;
        // You must override this to convert the page to the printer imaging model.
        virtual void    HandleEndPage( );
    protected:
        TPrintDevice( );
};
```

TPrintDevice converts a document to the printer imaging model. It provides an abstract interface to access page and job information and to process the job (a Print Run) and each page. Subclasses implement HandleRenderPage method to convert the page data to the printer imaging model. TPrintDevice is one of the objects that will be included in the personality document that the developer can edit or subclass. Therefore, it is possible for the developer to supply an implementation for converting the document to the printer imaging model. TPrintDevice is resurrected from the printer personality by the imaging task. RenderPrintRun is called with a TPrintRun, TPrintJobDescription, and a TPrinterIdentity. RenderPrintRun calls HandleBeginPrintRun, HandleRenderPrintRun, and HandleEndPrintRun. The reason for providing the begin and end methods is so that the subclasses can send some global commands to the printer before and after the document is processed. The default implementation of HandlePrintRun goes through the printrun in forward order and calls RenderPage for each page. Subclasses can override this to "play" the document in any random order they like. RenderPage calls HandleBeginPage, HandleRenderPage and HandleEndPage. Again, the reason for providing the begin and end methods is so subclasses can send page level global commands to the printer.

The TPrintJobDescription parameter, passed in the constructor, gives the user selected print time options. The print device maps the user's choice to what is actually available on the printer. It uses the printer's configuration kept in the printer's persistent store (the printer identity object can be used to get to the persistent store.).

TPrinterConfiguration

```
class TPrinterConfiguration : public MCollectible {
    public:
        virtual~TPrinterConfiguration( );
        // for static state of the printer
        virtual TPageDescription&   GetDefaultPageDescription( ) const = 0;
        // subclasses can return static or current state of the printer from
            // following methods.
        virtual TPageList&          GetPageList( ) const = 0;
        virtual TMediaList&         GetMediaList( ) const = 0;
        virtual TResolutionList&    GetResolutionList( ) const = 0;
        virtual TRGBColor           GetEngineColor( ) const = 0;
        // for dynamic state of the printer
        virtual TPageDescription&   GetCurrentPageDescription( ) const = 0;
        virtual void                SetPageList(TpageList&) = 0;
        virtual void                SetMediaList(TMediaList&) = 0;
        virtual void                SetResolution(TResolution&) = 0;
        virtual TStream&    operator<<=(Tstream& fromwhere);
        virtual TStream&    operator>>=(TStream& towhere) const;
        virtual Boolean     IsEqual(const MCollectible*) const;
        virtual long        Hash( ) const;
```

-continued

TPrinterConfiguration

```
protected:
    TPrinterConfiguration( );
    TPrinterConfiguration&operator=(const TPrinterConfiguration&);
};
```

TPrinterConfiguration is an abstract base class for a printer's state. It is part of the printer's persistent data kept current by the printer handler. Subclasses can store the actual configuration data in a file of their choice (e.g. dictionary, flat stream, etc.). For example, TPSPrinterConfiguration will use PPD files converted to a disk dictionary to keep configuration data. TPrinterConfiguration defines a protocol that provides for setting and getting configuration items such as page sizes, media, resolution(s), etc. When a printer is first available for use, its persistent store (a TModelStore) is created and TPrinterConfiguration is asked to write itself into the store. This becomes the printer's initial state which is updated when a print job is sent to it.

The lists returned by getters (TPageList, TMediaList, etc.) are implemented using C++ templates. As mentioned earlier, each type of printer has a subclass of TPrinterConfiguration that returns the printer's static configuration. This is streamed into the personality document which is given to the developer of that type of printer. The developer typically will edit the configuration instance (that is, the fields of the particular TPrinterConfiguration class) to include the printer's data.

TPrintJobDescription

```
class TPrintJobDescription : public MCollectible
{
    public:
        MCollectibleDeclarationsMacro(TPrintJobDescription);
    public:
                                TPrintJobDescription(TPrinterIdentity&);
                                TPrintJobDescription(const
                                TPrintJobDescription&);
        virtual ~TPrintJobDescription( );
    typedef enum EBannerPage { eNoBanner, eBriefBanner, eWholeBanner };
        //---------------------------------
        // Description:  These member functions are pretty much self explanatory
                         except for notes as added.
        // Requires    :
        // Modifies    : The Job's state is altered to reflect the requested
        //               operation.
        // Effects     :
        // Raises      :
        // Override    : All subclasses must override all virtual functions.
        //---------------------------------
        virtual void            SetCopies(unsigned long);
        virtual unsigned long       GetCopies( ) const;
        virtual unsigned long       GetpageCount( ) const;
        virtual void            SetPageCount(unsigned long);
        virtual void            SetCoverPage(EBannerpage);
        virtual EBannerpage     GetCoverPage( ) const;
        virtual void            SetEndPage(EBannerpage);
        virtual EBannerPage     GetEndPage( ) const;
        //---------------------------------
        // Punt choice specifies what the user wants to do when the page sizes
        // don't match between the document and the printer. These options
        // are defined in PageDescription.h
        //---------------------------------
        virtual void            SetJobPuntChoice(EPuntOption choice);
        virtual EPuntOption     GetJobPuntChoice( ) const;
        // Get the printer that this PrintJobDescription comes from
        virtual void            GetPrinterIdentity(TPrinterIdentity&) const;
        // User interface. Equivalent to the classic Print Job dialog.
        virtual void            EditSelf( );
        // MCollectible support
        virtual TStream&                operator<<=(TStream& fromwhere);
        virtual TStream&                operator>>=(TStream& towhere) const;
        virtual TPrintJobDescription&   operator=(const TPrintJobDescription&);
        virtual Boolean         operator==(const TPrintJobDescription&) const;
        virtual Boolean                 IsEqual(const MCollectible* right) const;
    protected:
                        TPrintJobDescription( );   // for Streaming
};
```

TPrintJobDescription is a base class providing protocol for accessing/changing print time options like number of copies, cover page choices, punt options (what to do when

there is a mismatch between the document and printer page sizes), etc. Developers can subclass this to add features specific to their device. The default implementation of TPrintJobDescription provides the choice of options common to all printers. The print job description gets streamed with a print job and is used by TPrintDevice (in the printer handler task) to send appropriate commands to the printer that implement the print time options. Each job description knows the printer that it is comes from. As a matter of fact, it is created by the printer model. The printer model gets it out of the printer's personality document. EditSelf is a place holder for a method that might be used to allow users to change job description attributes. This method might be called in response to the system wide print menu command.

While the invention is de scribed in terms of preferred embodiments in a specific system environment, those skilled in the art will recognize that the invention can be practiced, with modification, in other and different hardware and software environments within the spirit and scope of the appended claims.

Having thus described our invention, what I claim as new, and desire to secure by Letters Patent is:

1. Apparatus for use on a computer system having an imaging device, which is controlled by command primitives and pixelmaps and has a complexity limit, and a storage, the apparatus imaging a complex graphic image comprised of a plurality of graphic objects stored in the storage and comprising:

means responsive to each of the plurality of graphic objects for identifying imageable objects and non-imageable objects;

means for determining a complexity of each non-imageable object;

means for decomposing each non-imageable object with a complexity less than a predetermined threshold which depends on the imaging device complexity limit into a plurality of imageable objects;

means for converting each imageable object directly into command primitives;

means for transforming each non-imageable object with a complexity greater than the predetermined imaging device complexity threshold into a pixelmap; and

means for forming command primitives and pixelmaps into an image data stream to control the imaging device.

2. The apparatus described in claim 1 wherein the decomposing means comprises means for converting the non-imageable object to a plurality of imageable line segments.

3. The apparatus described in claim 1 wherein the determining means comprises means for adjusting a complexity of a non-imageable object based upon an intersection of the non-imageable object with other imageable objects.

4. The apparatus described in claim 1 wherein the transforming means comprises:

means for generating a pixelmap of the non-imageable object; and

means for rendering any intersections between the non-imageable object and an imageable object into said pixelmap.

5. The apparatus described in claim 4 wherein the generating means comprises:

means for generating a rectangular pixelmap enclosing the non-imageable object; and

means for establishing a clipping region surrounding the non-imageable object within the rectangular pixelmap.

6. The apparatus described in claim 5 wherein the rendering means comprises:

means for rendering any intersections between the rectangular pixelmap and a imageable object into the rectangular pixelmap; and

means for rendering a composite image of any intersections between the clipping region and a imageable object into the rectangular pixelmap.

7. The apparatus described in claim 1 wherein said imaging device is a printer.

8. The apparatus described in claim 1 wherein said imaging device is a display device.

9. A computer program product for use on a computer system having an imaging device, which is controlled by command primitives and pixelmaps and has a complexity limit, and a storage, the computer program product imaging a complex graphic image comprised of a plurality of graphic objects stored in the storage and comprising a computer usable medium having computer readable program code thereon including:

program code responsive to each of the plurality of graphic objects for identifying imageable objects and non-imageable objects;

program code for determining a complexity of each non-imageable object;

program code for decomposing each non-imageable object with a complexity less than a predetermined threshold which depends on the imaging device complexity limit into a plurality of imageable objects;

program code for converting each imageable object directly into command primitives;

program code for transforming each non-imageable object with a complexity greater than the predetermined imaging device complexity threshold into a pixelmap; and

program code for forming command primitives and pixelmaps into an image data stream to control the imaging device.

10. The computer program product described in claim 9 wherein the decomposing program code comprises program code for converting the non-imageable object to a plurality of imageable line segments.

11. The computer program product described in claim 9 wherein the determining program code comprises program code for adjusting a complexity of a non-imageable object based upon an intersection of the non-imageable object with other imageable objects.

12. The computer program product described in claim 9 wherein the transforming program code comprises:

program code for generating a pixelmap of the non-imageable object; and

program code for rendering any intersections between the non-imageable object and an imageable object into said pixelmap.

13. The computer program product described in claim 12 wherein the generating program code comprises:

program code for generating a rectangular pixelmap enclosing the non-imageable object; and

program code for establishing a clipping region surrounding the non-imageable object within the rectangular pixelmap.

14. The computer program product described in claim 13 wherein the rendering program code comprises:

program code for rendering any intersections between the rectangular pixelmap and a imageable object into the rectangular pixelmap; and

program code for rendering a composite image of any intersections between the clipping region and a imageable object into the rectangular pixelmap.

15. A method for use on a computer system having an imaging device which is controlled by command primitives and pixelmaps and has a complexity limit, and a storage, the method imaging a complex graphic image comprised of a plurality of graphic objects stored in the storage and comprising the steps of:

(a) identifying imageable objects and non-imageable objects in the plurality of graphic objects;

(b) determining a complexity of each non-imageable object;

(c) decomposing each non-imageable object with a complexity less than a predetermined threshold which depends on the imaging device complexity limit into a plurality of imageable objects;

(d) converting each imageable object directly into command primitives;

(e) transforming each non-imageable object with a complexity greater than the predetermined imaging device complexity threshold into a pixelmap; and

(f) forming command primitives and pixelmaps into an image data stream to control the imaging device.

16. The method described in claim 15 wherein step (c) comprises the step of:

(c1) converting the non-imageable object to a plurality of imageable line segments.

17. The method described in claim 15 wherein step (b) comprises the step of:

(b1) adjusting a complexity of a non-imageable object based upon an intersection of the non-imageable object with other imageable objects.

18. The method described in claim 15 wherein step (e) comprises the steps of:

(e1) generating a pixelmap of the non-imageable object; and

(e2) rendering any intersections between the non-imageable object and an imageable object into said pixelmap.

19. The method described in claim 18 wherein step (e1) comprises the steps of:

(e1a) generating a rectangular pixelmap enclosing the non-imageable object; and

(e1b) establishing a clipping region surrounding the non-imageable object within the rectangular pixelmap.

20. The method described in claim 19 wherein step (e2) comprises the steps of:

(e2a) rendering any intersections between the rectangular pixelmap and a imageable object into the rectangular pixelmap; and

(e2b) rendering a composite image of any intersections between the clipping region and a imageable object into the rectangular pixelmap.

21. The method described in claim 15 wherein said imaging device is a printer.

22. The method described in claim 15 wherein said imaging device is a display device.

* * * * *

US005768564A

United States Patent [19]

Andrews et al.

[11] Patent Number: 5,768,564

[45] Date of Patent: Jun. 16, 1998

[54] **METHOD AND APPARATUS FOR TRANSLATING SOURCE CODE FROM ONE HIGH-LEVEL COMPUTER LANGUAGE TO ANOTHER**

[75] Inventors: **Kristy A. Andrews**, Palo Alto; **Paul Del Vigna**; **Mark E. Molloy**, both of San Jose, all of Calif.

[73] Assignee: **Tandem Computers Incorporated**, Del.

[21] Appl. No.: **319,682**

[22] Filed: **Oct. 7, 1994**

[51] **Int. Cl.**[6] **G06F 9/44**

[52] **U.S. Cl.** **395/500**; 395/650; 395/685; 395/701; 395/705; 395/707; 395/708; 707/101; 364/280.4; 364/DIG. 1

[58] **Field of Search** 395/500, 700, 395/600, 708, 421.08, 800, 421.04, 611, 613, 650, 601, 680, 685, 701, 702, 703, 705, 707; 707/101, 103

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,599,691	7/1986	Sakaki et al.	364/419
4,667,290	5/1987	Goss et al.	395/700
4,794,528	12/1988	Hirose et al.	364/300
4,868,743	9/1989	Nishio	304/200
4,956,809	9/1990	George et al.	395/600
4,989,132	1/1991	Mellander et al.	395/700
5,151,697	9/1992	Bunton	341/51
5,173,853	12/1992	Kelly et al.	364/419
5,239,663	8/1993	Faudemay et al.	395/800
5,276,874	1/1994	Thomson	395/600
5,367,683	11/1994	Brett	395/700
5,384,568	1/1995	Grinberg et al.	341/51
5,392,390	2/1995	Crozier	395/144

FOREIGN PATENT DOCUMENTS

0 361 737 A3	4/1990	European Pat. Off.	G06F 9/44
0 583 117 A2	2/1994	European Pat. Off.	9/44

OTHER PUBLICATIONS

"Project Plan for pTAL to C++ Translator," Jul. 6, 1992, Reasoning Systems Inc.

Platoff, Michael, et al. "An Integrated Program Representation and Toolkit for the Maintenance of C Programs", *Proceedings, Conference on Software Maintenance Sorrento*, Oct. 15–17, 1991, No. Oct. 15, 1991, Institute of Electrical and Electronics Engineers, 129–137.

Ladd, D.A., et al. "A*: a Language for Implementing Language Processors", *Proceedings of the International Conference on Computer Languages, Toulouse*, May 16–19, 1994, No. Conf. 5, May 16, 1994, Institute of Electrical and Electronics Engineers, 1–10.

Albrecht, P., et al., "Source-to-Source Translation: Ada to Pascal and Pascal to Ada," *Proc. of the ACM Sigplan Symp. on the Ada Prog. Lang., Sigplan Notices* (Nov. 15, 1980) ACM 0-89791-0303/80/1200/0183, pp. 1–12.

Atkinson, R., et al., "Experiences Creating a Portable Cedar," Xerox PARC, CSL-89-8 (Jun. 1989), ©1989 Assoc. of Computing Machinery, pp. 1–12.

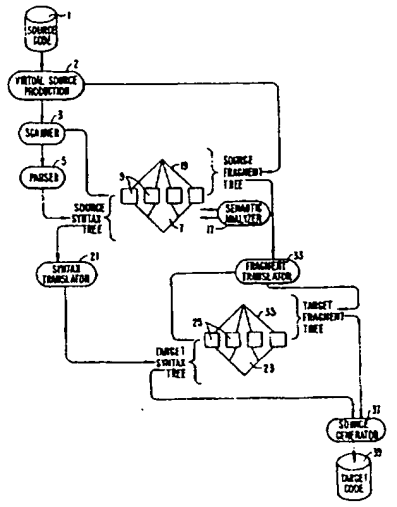
(List continued on next page.)

Primary Examiner—Kevin J. Teska
Assistant Examiner—Thai Phan
Attorney, Agent, or Firm—Townsend and Townsend and Crew LLP

[57] **ABSTRACT**

A method, system, apparatus, and program for translating one computer language to another using doubly-rooted tree data structures. A doubly-rooted tree is the combination of two sets of hierarchically related objects sharing a common set of leaves. An N-rooted tree is also described. When a doubly-rooted tree is constructed in the specified manner and then translated to a second doubly-rooted tree, source language code is transformed into target language code. In addition, the translation preserves preprocessor characteristics of the source language code including macros, conditionally compiled regions of code, source inclusion statements, and comments.

3 Claims, 9 Drawing Sheets



OTHER PUBLICATIONS

Bartlett, J., "Scheme -7C a Portable Scheme-to-C Compiler." ©Digital Equipment Corp. (1989), pp. 1-25.
Lemkin, P., Psail: Sail to C. *Computer Language* (Aug. 1985), pp. 39-45.
Moynihan, V., et al., "The Design and Implementation of a High-Level Language Converter," *Software Practice and Experience*, vol. 21 (4) (Apr. 1991), ©1991. J. Wiley & Sons, pp. 391-400.
Waters, R., "Program Translation via Abstraction and Reimplementation," *IEEE Translations on Software Engineering*, vol. 14, (8) (Aug. 1988), pp. 1207-1228.
Weiner, J., et al., "A Piggy-back Compiler for Prolog." ©1988 ACM 0-89791-269-1/88/0006/0288, pp. 288-296.
"SA-Sculptor/TAL TAL to C Conversion from Tandem TAL to ANSI C." prepared by Systems Automation, Inc., ©Systems Automation, Inc. (1992-93), p. 10.
"Recommendations for TAL to Tandem C Translation." prepared by Software Engineering and Enhancement Center for Tandem Computers, Inc. (Dec. 7, 1989), p. 55.
Project Plan for pTAL to C++ Translator. ©Tandem Computers Inc. (1992), 64 pages.
Minor, S., "Interacting with Structured-Oriented Editors." *Inter. J. of Man-Machine Studies* (1992) 37, pp. 299-418.
Ohsawa, Y., et al., "A new tree type data structure with homogeneous nodes suitable for a very large spacial database." *Data Eng.* (1990), 6th Inter. Conf., pp. 296-303.
Jiang, J., "Implementing the Patricia data structure for compression algorithms with finite-size dictionaries." *Data Transmission*, IEEE Conf., pp. 123-127.
Softfare II, a second conference on software development tools, techniques and alternatives (cat No. 85ch2231-9) SF, Dec. 1995 ISBN 0-89791-173-3, 1985, Washington, D.C., IEEE Computer Soc. Press USA, pp. 199-206.
Wang, et al., "A Tool for Tree Pattern Matching", *Proc. of the 1991 IEEE Int. Conf. on Tools for Artificial Intelligence*, TAI '91, (Cat 91CH3054-4) San Jose, Calif., Nov. 1991 ISBN 0-8186-2300-4, 1991 pp. 436-444 XP 000333437.
Micallef, et al., "Support for Algorithms for incremental attribute evaluation of asynchronous subtree replacements." IEEE transactions on software engineering, vol. 19, No. 3, Mar. 1993 NY, pp. 231-252 XP 000382081.
Cardenas, et al., "Doubly-chained tree data base organization-analysis and design strategies." *Computer Journal*, vol. 20, No. 1, Feb. 1977 ISSN, pp. 15-26.

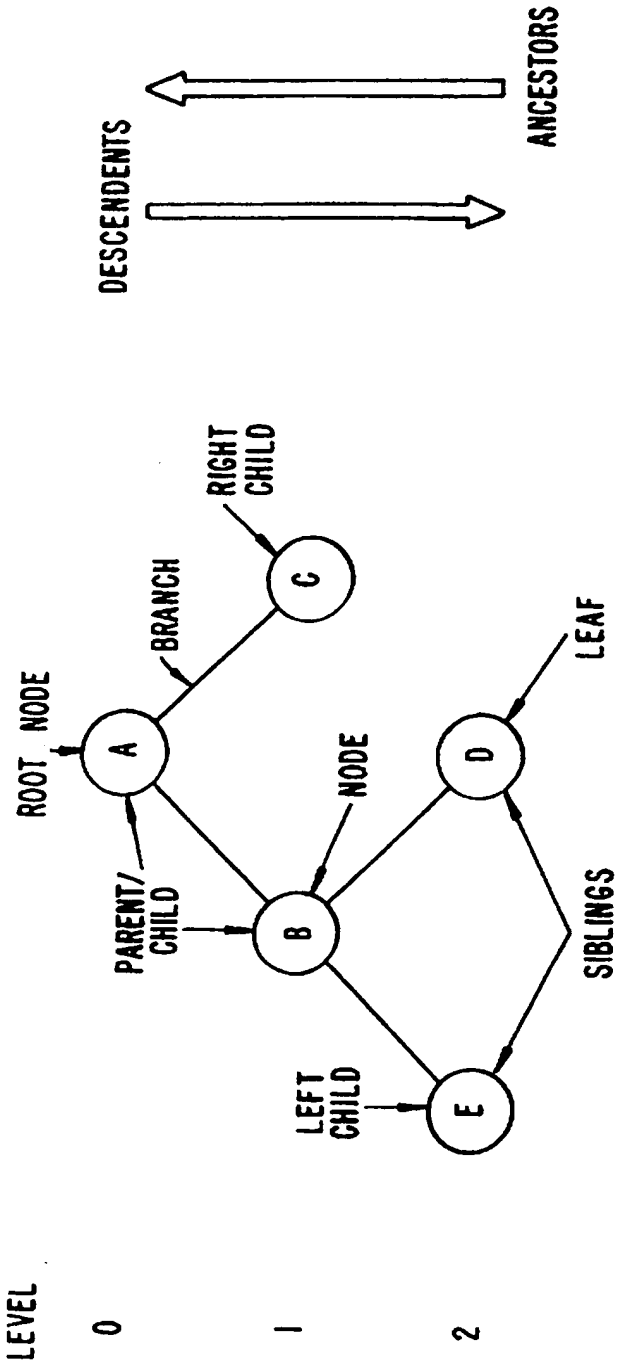

DESCENDENTS
ANCESTORS
ROOT NODE
BRANCH
RIGHT CHILD
PARENT/ CHILD
NODE
LEAF
SIBLINGS
LEFT CHILD
A
B
C
D
E
LEVEL
0
1
2

FIG. 1.

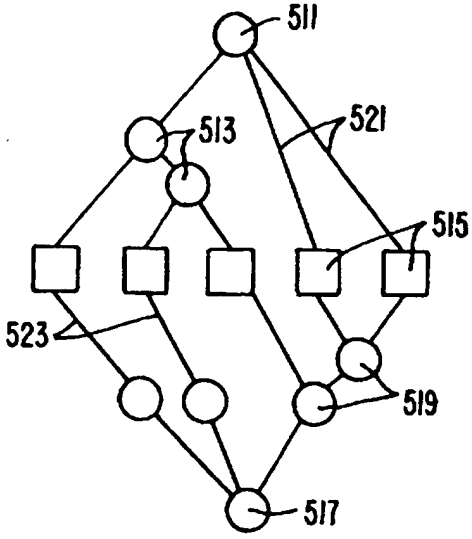

511
513
521
515
523
519
517

FIG. 2.

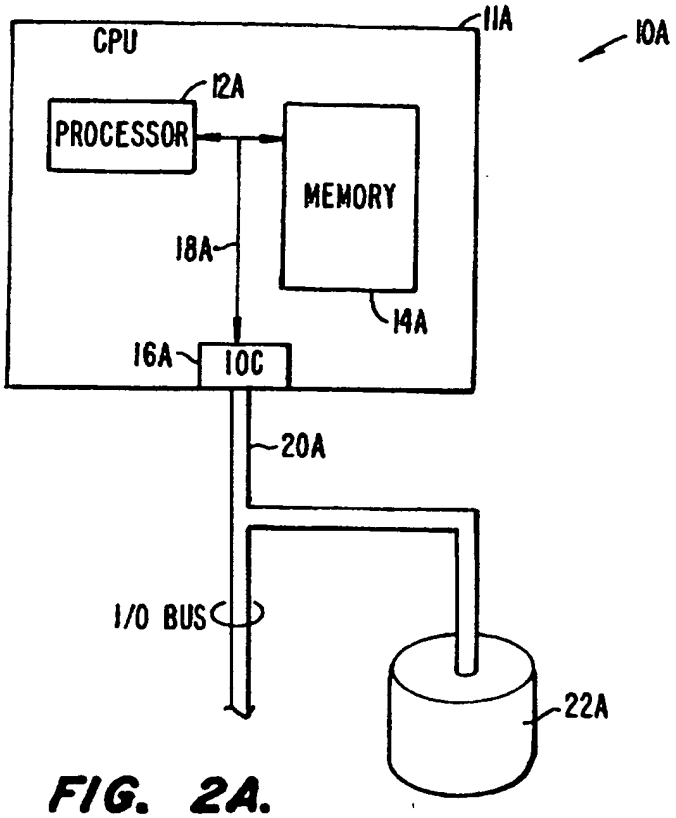

CPU
11A
10A
12A
PROCESSOR
MEMORY
18A
14A
16A
10C
20A
I/O BUS
22A

FIG. 2A.

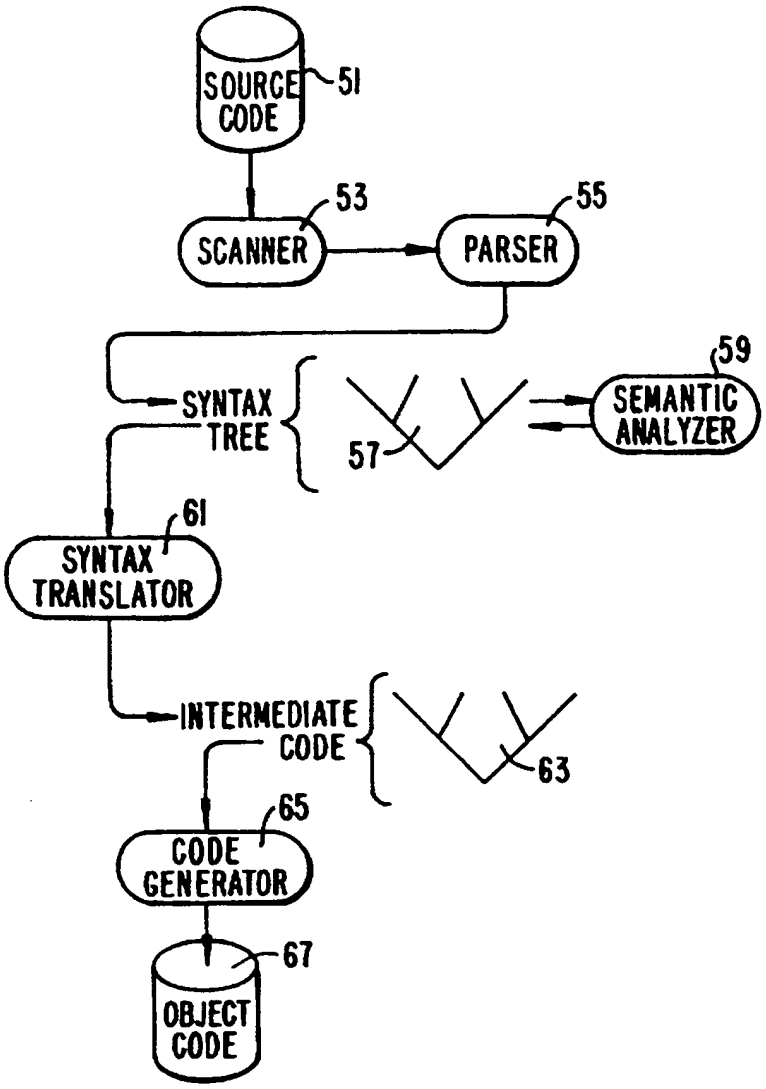


FIG. 3.
PRIOR ART

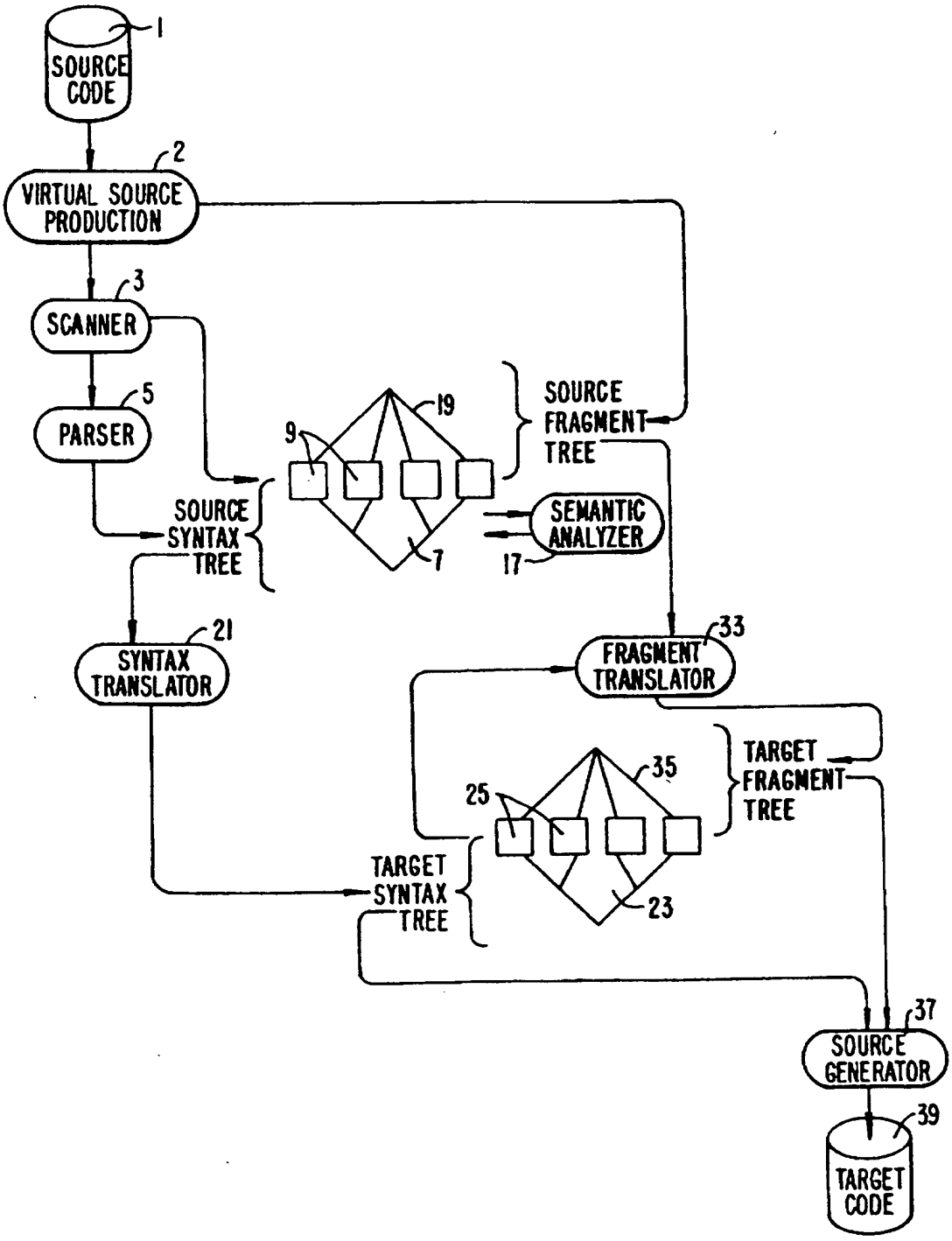

SOURCE CODE
1
VIRTUAL SOURCE PRODUCTION
2
SCANNER
3
PARSER
5
SOURCE SYNTAX TREE
9
19
7
SOURCE FRAGMENT TREE
SEMANTIC ANALYZER
17
SYNTAX TRANSLATOR
21
FRAGMENT TRANSLATOR
33
TARGET FRAGMENT TREE
35
25
23
TARGET SYNTAX TREE
SOURCE GENERATOR
37
TARGET CODE
39

FIG. 4.

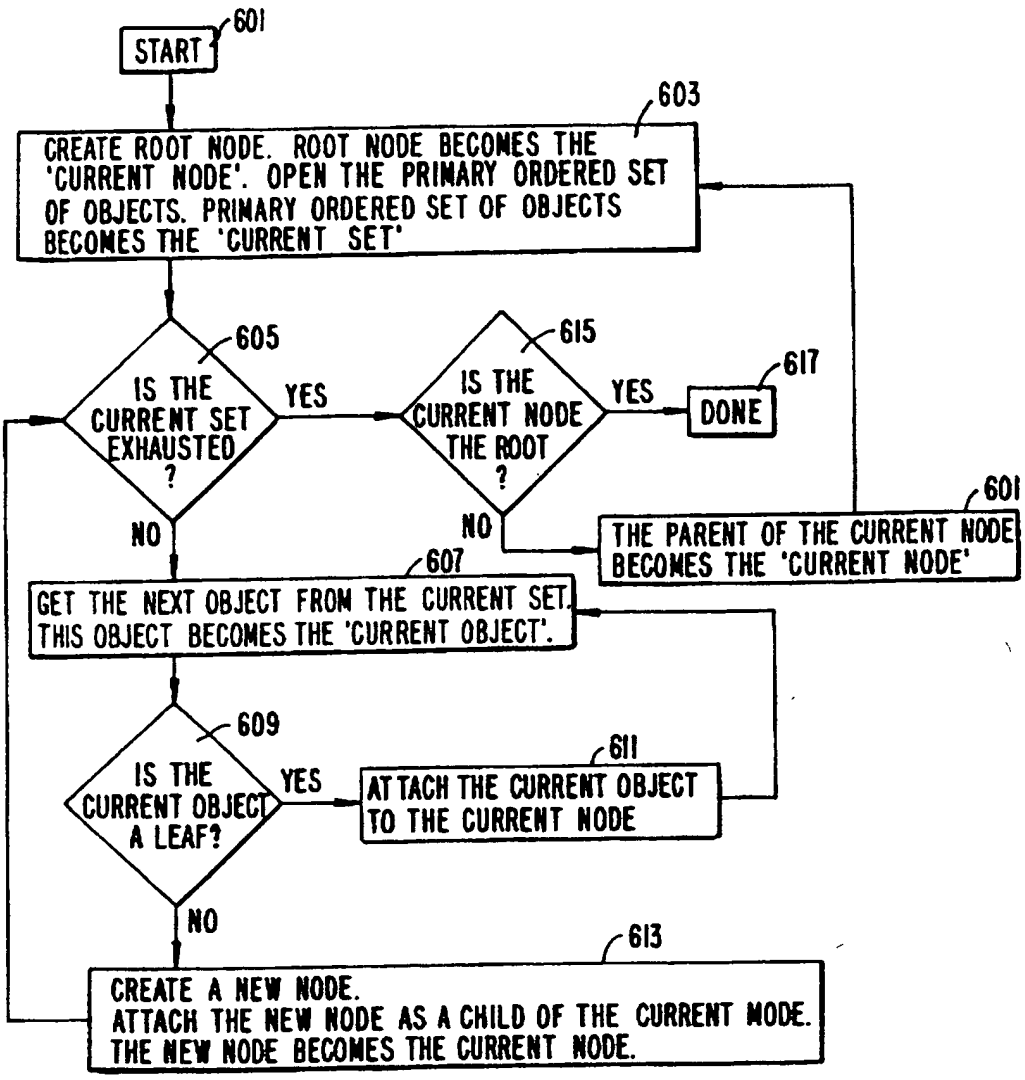

START
601
603
CREATE ROOT NODE. ROOT NODE BECOMES THE 'CURRENT NODE'. OPEN THE PRIMARY ORDERED SET OF OBJECTS. PRIMARY ORDERED SET OF OBJECTS BECOMES THE 'CURRENT SET'
605
IS THE CURRENT SET EXHAUSTED ?
YES
615
IS THE CURRENT NODE THE ROOT ?
YES
617
DONE
NO
NO
601
THE PARENT OF THE CURRENT NODE BECOMES THE 'CURRENT NODE'
607
GET THE NEXT OBJECT FROM THE CURRENT SET. THIS OBJECT BECOMES THE 'CURRENT OBJECT'.
609
IS THE CURRENT OBJECT A LEAF?
YES
611
ATTACH THE CURRENT OBJECT TO THE CURRENT NODE
NO
613
CREATE A NEW NODE.
ATTACH THE NEW NODE AS A CHILD OF THE CURRENT NODE.
THE NEW NODE BECOMES THE CURRENT NODE.

FIG. 5.

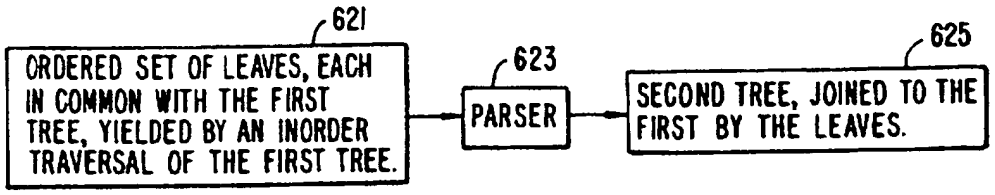

621
ORDERED SET OF LEAVES, EACH IN COMMON WITH THE FIRST TREE, YIELDED BY AN INORDER TRAVERSAL OF THE FIRST TREE.
623
PARSER
625
SECOND TREE, JOINED TO THE FIRST BY THE LEAVES.

FIG. 6.

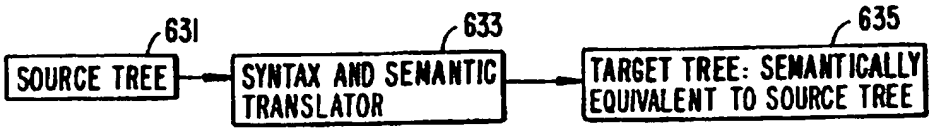

631
SOURCE TREE
633
SYNTAX AND SEMANTIC TRANSLATOR
635
TARGET TREE: SEMANTICALLY EQUIVALENT TO SOURCE TREE

FIG. 7.

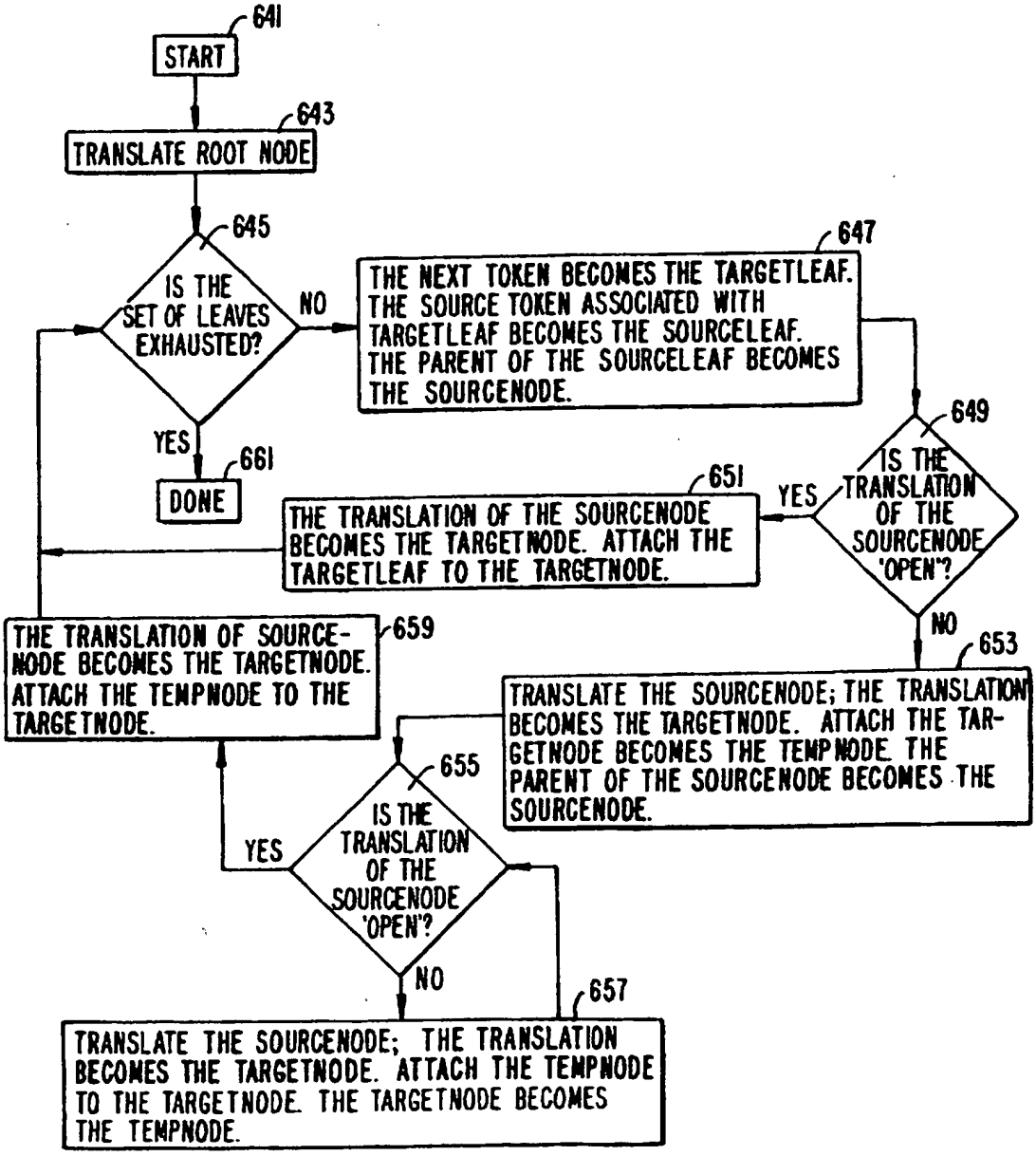

641
START
643
TRANSLATE ROOT NODE
645
IS THE SET OF LEAVES EXHAUSTED?
NO
647
THE NEXT TOKEN BECOMES THE TARGETLEAF. THE SOURCE TOKEN ASSOCIATED WITH TARGETLEAF BECOMES THE SOURCELEAF. THE PARENT OF THE SOURCELEAF BECOMES THE SOURCENODE.
649
IS THE TRANSLATION OF THE SOURCENODE 'OPEN'?
YES
651
THE TRANSLATION OF THE SOURCENODE BECOMES THE TARGETNODE. ATTACH THE TARGETLEAF TO THE TARGETNODE.
NO
653
TRANSLATE THE SOURCENODE; THE TRANSLATION BECOMES THE TARGETNODE. ATTACH THE TARGETNODE BECOMES THE TEMPNODE. THE PARENT OF THE SOURCENODE BECOMES THE SOURCENODE.
YES
661
DONE
659
THE TRANSLATION OF SOURCENODE BECOMES THE TARGETNODE. ATTACH THE TEMPNODE TO THE TARGETNODE.
655
IS THE TRANSLATION OF THE SOURCENODE 'OPEN'?
YES
NO
657
TRANSLATE THE SOURCENODE; THE TRANSLATION BECOMES THE TARGETNODE. ATTACH THE TEMPNODE TO THE TARGETNODE. THE TARGETNODE BECOMES THE TEMPNODE.

FIG. 8.

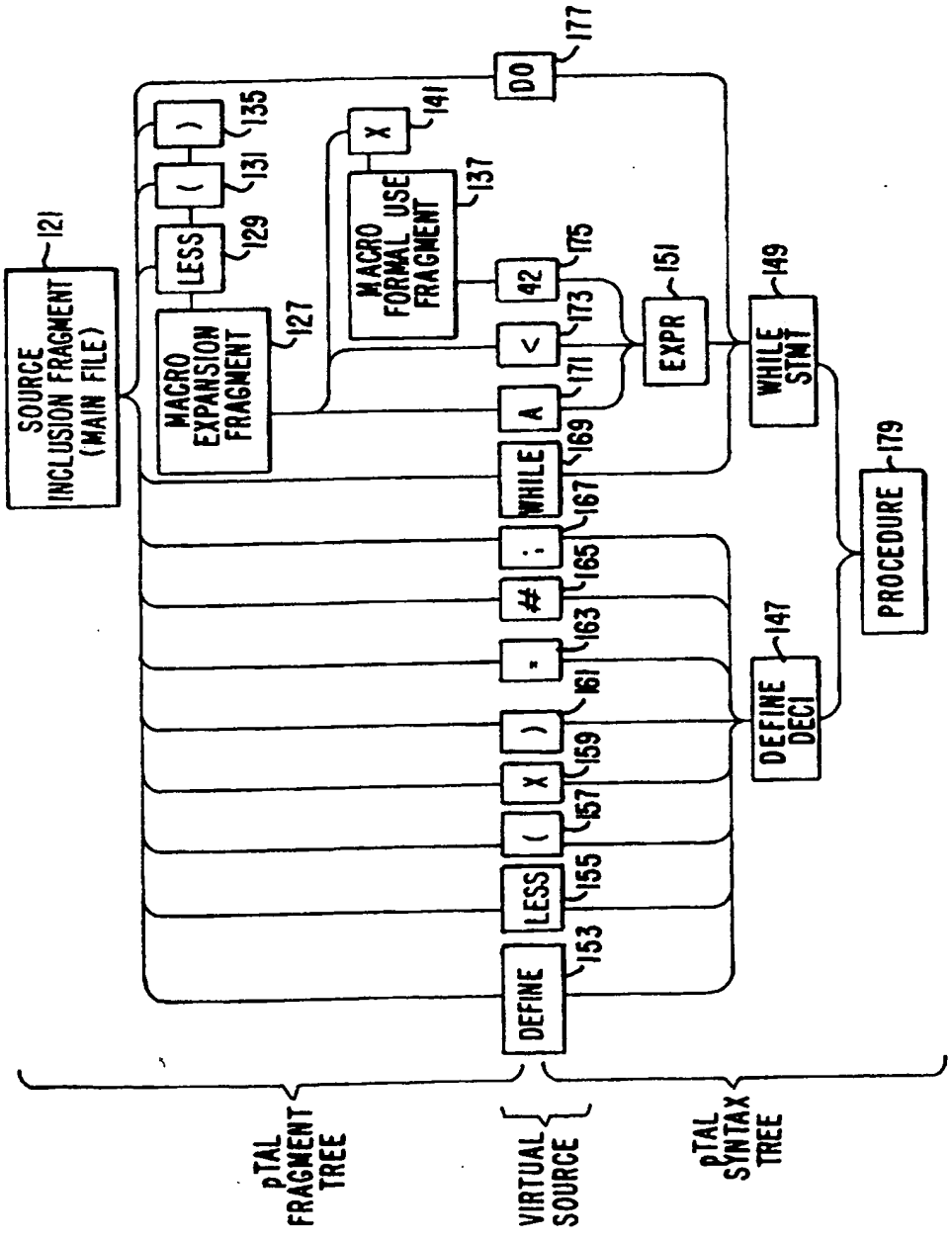

pTAL FRAGMENT TREE
VIRTUAL SOURCE
pTAL SYNTAX TREE
SOURCE INCLUSION FRAGMENT (MAIN FILE)
121
MACRO EXPANSION FRAGMENT
127
LESS
129
(
131
)
135
MACRO FORMAL USE FRAGMENT
137
X
141
DO
177
42
175
<
173
A
171
WHILE
169
:
167
#
165
-
163
)
161
X
159
(
157
LESS
155
DEFINE
153
EXPR
151
WHILE STMT
149
DEFINE DECL
147
PROCEDURE
179

FIG. 9.

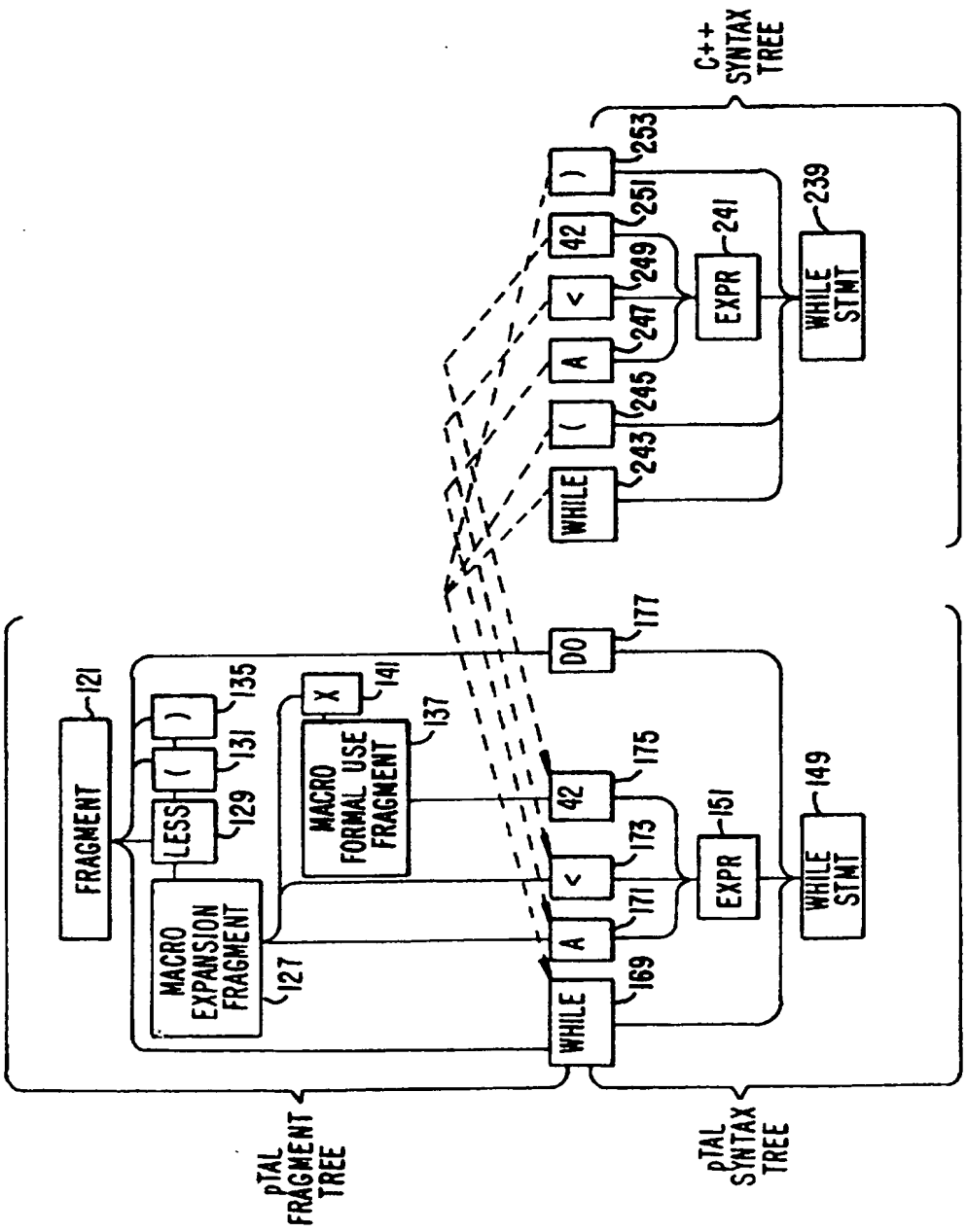

pTAL FRAGMENT TREE
FRAGMENT
121
LESS
129
(
131
)
135
MACRO EXPANSION FRAGMENT
127
MACRO FORMAL USE FRAGMENT
137
X
141
DO
177
WHILE
169
A
171
<
173
42
175
EXPR
151
WHILE STMT
149
pTAL SYNTAX TREE
C++ SYNTAX TREE
WHILE
243
(
245
A
247
<
249
42
251
)
253
EXPR
241
WHILE STMT
239

FIG. 10.

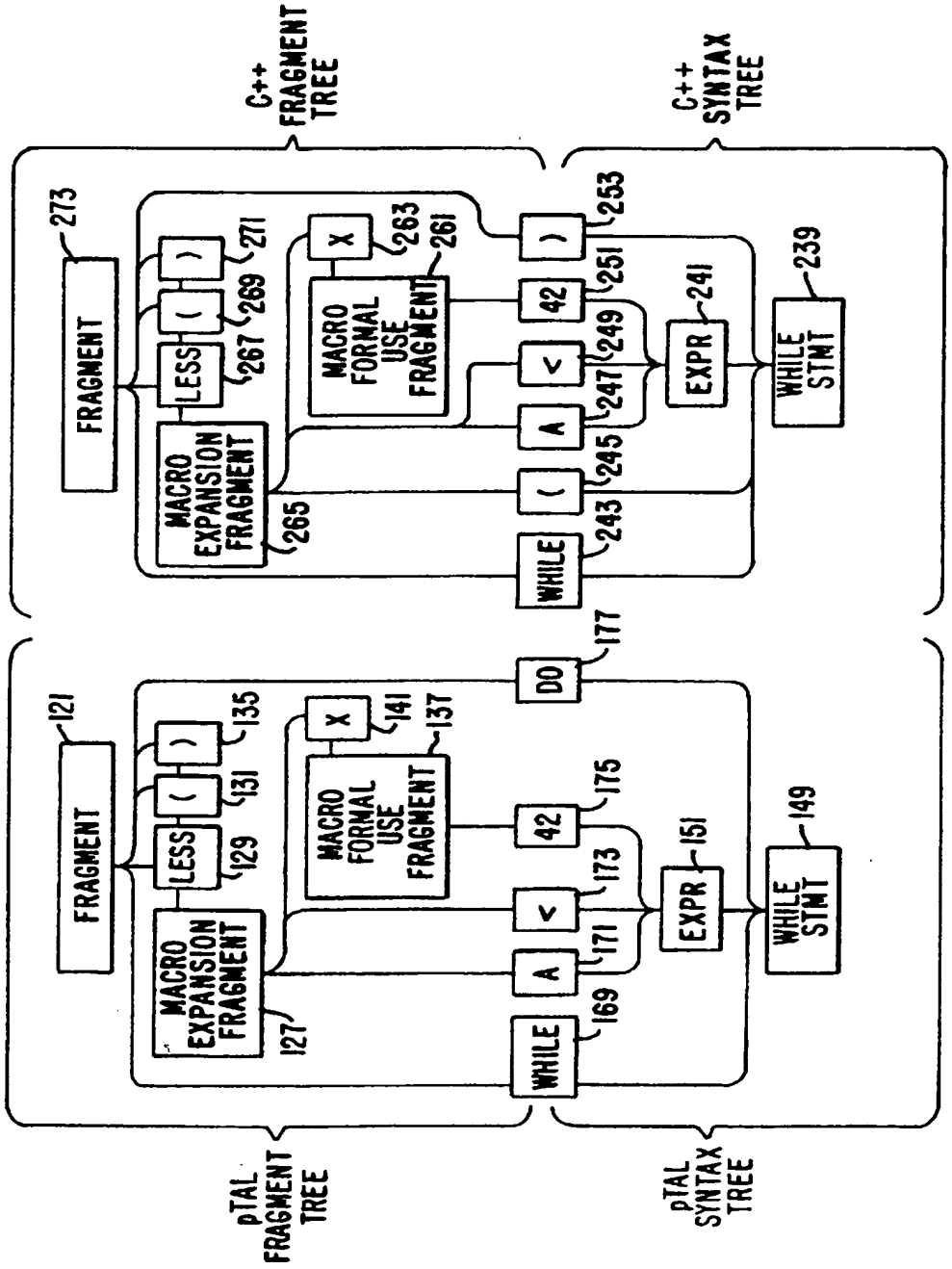

C++ FRAGMENT TREE
C++ SYNTAX TREE
pTAL FRAGMENT TREE
pTAL SYNTAX TREE
FRAGMENT
LESS
(
)
MACRO EXPANSION FRAGMENT
MACRO FORNAL USE FRAGMENT
X
WHILE
A
<
42
EXPR
WHILE STMT
DO
121
129
131
135
137
141
169
171
173
175
177
149
151
239
241
243
245
247
249
251
253
261
263
265
267
269
271
273

FIG. 11.

METHOD AND APPARATUS FOR TRANSLATING SOURCE CODE FROM ONE HIGH-LEVEL COMPUTER LANGUAGE TO ANOTHER

BACKGROUND OF THE INVENTION

This invention relates to automated translation between high-level computer programming languages.

This invention relates particularly to improved preservation in a target high-level language of preprocessor characteristics (such as macros, source file inclusion structure, and commentary) contained in a source high-level language. A feature of this invention is that preprocessor characteristics need not necessarily be processed by a preprocessor.

High-level computer languages enable computer programmers to communicate with computers. Statements programmers write in a computer language form a computer program which in turn instructs a computer to perform a set of tasks. "Compilation" is the manner in which high-level computer language programs are converted into instructions, generally called machine code, which the computer can understand and execute. A compiler is a computer program which performs this translation.

In general, each brand of computer understands a different set of machine code instructions. Therefore, a different compiler must exist for each computer to translate a high-level computer language. Because compilers for every high-level computer language do not exist on every brand of computer, not every program can execute on every machine. Programmers can only write programs in the languages for which compilers exist for their target computers.

Nonetheless, it is highly desirable to have a single computer program run on as many brands of computers as possible. Application programs are typically complex and difficult to write; rewriting programs in multiple languages to run on multiple brands of computers is impractical. Likewise, compilers are difficult to write; providing them for every language for every brand of computer is equally impractical. One way of addressing these problems has been the development of well known, widely used, standardized high-level languages. Compilers for these languages are available for a wide variety of computers.

The development of standardized languages has not been a complete solution. There exist numerous high-level languages, and many large programs written in them, which are exotic, highly specialized, little used, obsolete, or designed for specific computers. Many computers do not have compilers available for these languages.

Because many high-level computer languages, whether or not they are standardized, cannot be compiled on every computer, programs have to be translated to other languages. While translation can be done by hand, it is a laborious, time consuming, and expensive process prone to error. To address this problem, automatic translators have been and continue to be developed to translate programs written in one high-level language to another.

Automatic translators may be used in either of two distinct strategies to solve the problem of an unavailable compiler for a particular language on a particular computer. First, programmers may continue to write and maintain programs in the original source language. The translator converts these programs into intermediate code in a target language. An available compiler for the target language then converts this intermediate code into machine code which the target computer can understand. Although the target language is usually a standard widely available language, the translator does not have to produce readable or maintainable source code.

The second strategy requires a translator to produce readable and maintainable code. Programmers going this route want to abandon the original language in favor of the target. Building this type of translator is a more difficult task and is the focus of this invention.

Prior art attempts to build translators which produce readable code have had differing goals and various levels of success. Syntax of one high-level language has been successfully transformed into syntax of another high-level language. Some translators have produced attractively formatted target code. While source code comments have been migrated to target code, their placement has not always been optimal. Translators have also attempted to transform the style of programs to make them more readable. Others have used knowledge-based systems to extract the meaning of the source program and rewrite it in the target language.

However, prior art translators have universally failed adequately to preserve programming constructs generally known as preprocessor characteristics. Many high-level languages include a preprocessor language separate from but coexisting with the language itself. Characteristics (which are also herein referred to as invocation expressions) of the preprocessor language may include a conditional compilation mechanism, a macro mechanism, a source inclusion mechanism, a variety of compiler directives, and a comment mechanism. At the risk of oversimplification, the preprocessor allows programmers to use shorthand expressions for longer constructs. Thus, invoking the shorthand expression triggers a text substitution when the source code is run through the preprocessor.

The failure of translators to handle adequately preprocessor characteristics is a well known problem. Experts in the field, when considering how to replace a source language macro definition with a target language macro definition, have stated: "We do not know of a general, fully automatic mechanism for achieving this replacement." Two suggestions have been made:

- Attempt to parse macro bodies outside of their context of use, with the hope that some well-formed and complete source language constructs will translate them correctly.
- Recognize common sequences of tokens in macro bodies via pattern matching.

The former is not a general solution because languages may make no restrictions for using macros, and such restrictions, in any event, would make macros susceptible to semantic errors. The latter suggestion would only work in special cases.

SUMMARY OF THE INVENTION

A general object of the present invention is to provide a method and apparatus for automatically translating the source code from one high level computer language to the source code of another language.

A specific object of the present invention is to provide a method of translating code from a source computer language to a target while preserving the preprocessor characteristics of the code as written in the source language.

Another object of the present invention is to provide a data structure useful for automated translation, whether it be from one computer language to another or from a natural language to another. The data structure, termed a doubly-

rooted tree, combines two prior art tree data structures so that their leaves coincide. By creating and filling the doubly-rooted tree in a specified manner, and then translating it to a second doubly-rooted tree, preprocessor characteristics of a computer language may be preserved while translating other statements, constructs, and expressions.

A doubly-rooted tree may serve as the basis for more complex data structures. For example, additional trees may be joined at the leaves to form an N-rooted tree. Such a linked set of trees has computer database applications.

Other and further objects of the present invention will be apparent from the following description and claims and are illustrated in the accompanying drawings which, by way of illustration, show preferred embodiments of the present invention and the principles thereof and what are now considered to be the best mode contemplated for applying these principles. Other embodiments of the invention embodying the same or equivalent principles may be used and structural changes may be made as desired by those skilled in the art without departing from the present invention and the purview of the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

In the Drawings:

FIG. 1 depicts a prior art tree structure.

FIG. 2 depicts a doubly-rooted tree data structure, joined at the leaves, in accordance with the present invention.

FIG. 2A is a simplified block diagram of a computing system for use with the present invention;

FIG. 3 is a block diagram depicting data flow through a simple prior art compiler.

FIG. 3A is a block diagram depicting the syntax tree of FIG. 3 where the source language is FORTRAN and the source statment is $Z=X+Y$.

FIG. 4 is a block diagram depicting data flow through an automated source code translator in accordance with one embodiment of the present invention.

FIG. 4A is a block diagram depicting the source syntax tree and associated virtual source of FIG. 4 where the source language is FORTRAN and the source statement is $Z=X+Y$.

FIG. 5 is a flowchart describing a method for building a first tree in a doubly-rooted tree in accordance with one embodiment of the present invention. The method builds the tree from the top down in depth first traversal order.

FIG. 6 is a flowchart describing the building of a second tree in the doubly rooted tree introduced in FIG. 5, in accordance with one embodiment of the present invention. An inorder traversal of the first tree yields an ordered set of leaves that is input to the parser 623. The method builds the tree from the bottom up in depth first traversal order.

FIG. 7 is a flowchart describing the translation of a second tree in a first doubly rooted tree to a corresponding tree in a second doubly rooted tree in accordance with one embodiment of the present invention.

FIG. 8 is a flowchart describing the translation of a first tree in the first doubly rooted tree referred to in FIG. 5 to a corresponding tree in a second doubly rooted tree in accordance with one embodiment of the present invention.

FIG. 9 depicts an example fragment tree and syntax tree, using a high-level computer language, pTAL, as the source language.

FIG. 10 depicts the example PTAL source fragment tree and source syntax tree, including the leaves (tokens) that link them, introduced in FIG. 9. It also shows a target syntax tree of another high-level computer language, C++, that represents a translation of the source syntax tree and half of the target doubly-rooted tree.

FIG. 11 depicts the source doubly-rooted tree and the target doubly-rooted tree introduced in FIG. 10, after the translation of the source fragment tree. All four trees are illustrated: the example pTAL source fragment tree and source syntax tree as well as semantically equivalent C++ target fragment tree and target syntax tree.

DETAILED DESCRIPTION OF THE INVENTION

This invention implements a strategy for translating legacy software to a different high-level programming language. This translation process preserves text preprocessor mechanisms such as macros, conditionally compiled regions of code, and source inclusion. This section specifically discusses the design of the Rosetta Translator, which implements this strategy. The Rosetta Translator translates from Tandem's portable Transaction Application Language (pTAL) to C++.

Source-To-Source Translation

Source-to-source translation to a standard, portable, and widely-available language, such as C, has been shown to be an excellent means of making exotic languages available on many platforms [2, 3, 4, 9]. Programmers continue to write and maintain programs in the original source language, and use the translated source as intermediate code. These translators need not address the problems inherent in producing readable and maintainable code: preserving macro definitions and program structure, and retaining meaningful naming, comment placement, and format.

Attempts have been made to implement source-to-source translators that transform the style of a program, thus rendering it more readable, or to extract the meaning of a program and use a knowledge-based system to express the program in a higher-level language [8]. We were unable to apply such techniques to our advantage.

pTAL, C++, and many high-level programming languages consist of a preprocessor language separate from but coexisting with a high-level language. The preprocessor language might include a conditional compilation mechanism, a macro mechanism, a source inclusion mechanism, a variety of compiler directives, and a comment mechanism. The preprocessor language need not be implemented by a separate preprocessor. The Rosetta Translator not only translates high-level language constructs but also preprocessor constructs; preserving the latter is a difficult but important part of producing readable and maintainable code.

While some published accounts describe translators that produce readable high-level code [1, 5], we have encountered no published accounts of experience with translators which produce readable and maintainable code and which preserve preprocessor mechanisms.

Languages

The source language, pTAL, is a proprietary block-structured systems programming language, originally derived from Algol-60. The vast majority of Tandem's system code was written in pTAL or its precursor, TAL. The Rosetta Translator uses pTAL as its source language instead of TAL because pTAL excludes various architecture-specific TAL features that are not available in standard programming languages, such as assembly language code statements and

arbitrary register manipulation. All TAL code will eventually migrate to pTAL.

The translated code is dominated by the C language subset of C++, but includes occasional constructs specific to C++. The Rosetta Translator does not generate object-oriented code. C++ is a much better target language than C for a number of reasons, including strong type checking and easy emulation of several important pTAL language constructs, and the opportunity to use the object-oriented paradigm in the future. The ideas presented here apply to other programming languages containing preprocessor mechanisms that one would have a practical need to translate.

Concepts

The following concepts are key to this description of a method for preserving macros.

Virtual Source

A program's virtual source is a stream of tokens that a compiler parses to create a syntax tree representing that program.

Virtual Source Production Mechanisms

Virtual source production mechanisms are used by a scanner or preprocessor to create virtual source. Virtual source production mechanisms available in pTAL and C++ are:

source inclusion of files

macro expansion

macro formal parameter substitution

A virtual source production mechanism can appear anywhere in a source file and yield any portion of a syntactic construct; it need not honor syntactic construct boundaries. Virtual source does not necessarily exist in textual form in any single source file; the token stream is created by a series of source inclusions, macro expansions, and macro formal parameter substitutions.

The Rosetta Translator translates virtual source production mechanisms as well as high-level language constructs. It does not merely expand the virtual source with a preprocessor before performing its translation. The macro that the programmer knew and loved appears in the translated code, and means the same thing. The file structure of the program is also preserved.

Fragment

A "fragment" represents the result of an invocation of a virtual source production mechanism: source file inclusion, macro expansion, or macro formal parameter substitution. Each fragment is linked to the tokens that make up its invocation syntax: a macro formal parameter, a macro invocation (including the parameter list), or a token representing a source inclusion directive. Each fragment contains the tokens composing its expansion: a macro actual parameter, a macro body, or an included file. The tokens composing the invocation syntax are themselves contained in another fragment, which represents its context of use. For example, a macro invocation can be embedded in a macro body, or in the main program file.

Fragments can be divided into "partitions" by compiler directives. Because some compiler directives affect the semantics of the program, and some determine what text is scanned (for example, conditional compilation directives), directive boundaries must be preserved.

Partitions are bounded by:

compiler directives

beginning and end of macro bodies in a macro definition

beginning and end of a file

The details of preserving partition boundaries are not discussed in this document.

Fragment Tree

A fragment tree represents each source inclusion, macro expansion, and macro actual parameter substitution that was employed to create the virtual source.

Static Fragments

A static fragment is a mold from which fragments are made; each fragment is an instance of a static fragment. A source file is a static fragment, and it may contain nested static fragments. The Rosetta Translator pieces together textual representations of C++ fragments to form C++ output files, fitting a textual representation of a macro body, for example, into its definition. Because the Rosetta Translator's design requires that each source language macro maps to exactly one target language macro, it checks the textual consistency of every macro use, to ensure that the macro text "works" in all contexts of use. It checks the textual consistency between fragments generated during previous translation sessions and those generated during the "current" session, to ensure that pieces of an included file "work" in all contexts of use.

Translator Overview

This section introduces a "fragment tree" as a means of representing macro invocations, macro actual parameter substitutions, and source inclusions in a program. It gives an overview of the translator phases, including steps to preserve the fragment tree. It then describes in detail the structure of the fragment tree and gives a fragment tree translation method.

The Rosetta Translator behaves like a normal compiler, with a few additional steps. A pTAL syntax tree represents the syntactic structure of the virtual source. Tokens appear on the leaves of the otherwise traditional abstract syntax tree. Each token is also associated with a fragment, indicating which virtual source production mechanism brought it into the virtual source. The syntax tree and the fragment tree are said to be "joined at the tokens."

The scanning phase builds the pTAL fragment tree. The pTAL syntax tree is translated to a C++ syntax tree. An additional translation step converts the pTAL fragment tree to the C++ fragment tree.

After translation, each C++ macro invocation fragment or source inclusion fragment contains a sequence of C++ tokens that represents text belonging in that macro body or included file. The source generator pieces together translated instances to form a generated text file. Tokens and comments are formatted (assigned file positions) based on the syntax that contains them, taking cues from the formatting of the corresponding source language construct. FIG. 4 depicts a general diagram of the Rosetta Translator's data flow.

Following is some sample pTAL source code, and the translated C++ source code.

pTAL Code	Generated C++ Code
define less(x) = a < x#;	#define less(x) a < x
. . .	. . .
while less(42) do . . .	while (less(42)) . . .

FIG. 9 illustrates the fragment tree that represents the pTAL while loop from the previous example. The tokens in the virtual source are in both the fragment tree and the syntax tree.

Properties of Fragments

Fragments have some interesting properties:

A pTAL fragment is semantically equivalent to the C++ fragment to which it translates. For a macro to "work", it must expand to a meaningful token stream in every context of use in the generated code. For a macro to be readable and maintainable, it must be invoked for the same purposes in the generated code as in the source code; it must provide the same benefit by encapsulating equivalent target language text. These considerations for macros also apply to code included from another file and to macro actual parameters.

Tokens in a pTAL fragment must be translated in their context or contexts of use. Macros cannot be translated immediately at their point of definition because, even if the text could be parsed, semantic analysis would not be possible. Only when the macro is expanded in context is the semantic information necessary for translation available. Macros are, therefore, translated after their invocations have been expanded and analyzed. The actual parameters of a parameterized macro and regions of included text are processed analogously. The bodies of unused macros are not translated, and unused actual parameters are not translated.

An important implication of this property is that shared interface files containing macro definitions or conditional compilation tend to be translated incrementally, as a side effect of translating the modules that use them.

A pTAL fragment should translate to the same C++ text in every context of use. A macro body is translated once for each expansion in the virtual source. The Rosetta Translator chooses to produce exactly one target language macro for each source language macro; this design decision requires that the translation of every instance of a macro expansion be textually identical. The Rosetta Translator can only be sure that the translation of a macro "works" in every context of use if every instance of the macro expansion translates to textually identical generated code. A textual mismatch might occur, for example, when the syntax expressing two distinct operations is the same in the source language but different in the target language. The pun (one phrase with two distinct meanings) does not translate. Strategies for avoiding textual mismatches and for coping with mismatches that cannot be prevented are discussed later in this document. The same considerations apply to code included from another file and to macro actual parameters.

Construct translations should not blur fragment boundaries. Each pTAL syntactic construct is mapped to a corresponding C++ syntactic construct, and pTAL tokens within that construct are mapped to corresponding C++ tokens within the corresponding C++ construct. While the tokens in the resultant token stream need not maintain the same relative ordering as the pTAL tokens that generated them, fragment boundaries are points of ordering enforcement. Each token in a given C++ fragment is generated from some token in the same pTAL fragment. Strategies for avoiding fragment boundary violations and for coping with violations that cannot be prevented are discussed later in this document.

Translating the Syntax Tree

The Rosetta Translator translates the source language syntax tree, generating an equivalent target language syntax tree. The translation routines for most source language constructs generate idioms in the target language when appropriate.

The Rosetta Translator associates each generated target language token with one source language token. The "translation" of a source language token creates a target language token and establishes this association. The choice of which pTAL token is associated with a given C++ token determines the fragment in which that C++ token belongs: the C++ counterpart of the pTAL fragment in which the associated pTAL token resides.

Some constructs allow a one-to-one mapping from source language tokens to target language tokens. If tokens do not map one-to-one, the Rosetta Translator chooses token mappings that maximize "semantic congruence," that is, it tries to associate tokens with their logical counterparts. Choosing appropriate mappings helps ensure meaningful macro translations. The collection of tokens contained in a macro body (or macro actual parameter) in the source language tends to generate a collection with the same meaning in the target language.

FIG. **10** illustrates part of a pTAL syntax tree and fragment tree, joined by the pTAL tokens making up the virtual source. The dashed lines depict links between each C++ token and the pTAL token that generated it. The tokens in the relational expression map one-to-one from the source language to the target language. The source language "while" keyword generates the target language parentheses tokens in addition to the target language "while" token.

Translating the Fragment Tree

After building the C++ syntax tree, the Rosetta Translator traverses the target language virtual source token stream, and builds the C++ fragment tree, based on the position in the pTAL fragment tree of the source language token associated with each target language token encountered in the virtual source stream.

The method given in this section builds a target language fragment tree. Operations are as follows:

- Attach(x, y): Attach x to y: y becomes the parent of x. x is added positionally after pre-existing siblings.
- Translate(n): Translate n (token or fragment) to the target language version of the object, and return the translation.
- Translation(n): Return the latest translation of the fragment n; nil if it has not been translated.
- Parent(n): If n is a token, return the fragment in which n resides. If n is a fragment, return the invocation syntax of the fragment (macro name token, for example).
- IsOpen(n): Return true if the fragment n is open in the target fragment tree. The "open" nodes in the target language fragment tree include the ancestors of the last token that was added to the fragment tree. Adding a child to a fragment "closes" nodes that are no longer ancestors of the latest token added.

Begin by translating the main source include fragment. This fragment is initially open.

```
FOR (TargetToken = each token in the target language virtual source) BEGIN
        SourceToken = the source token associated with the current TargetToken
        SourceFrag = Parent (SourceToken)
        IF (IsOpen (Translation (SourceFrag)))
        THEN BEGIN
            /* add a new token to an existing fragment that is already open */
            TargetFrag = Translation (SourceFrag)
            Attach (TargetToken, TargetFrag)
        END ELSE BEGIN
            /* create a new fragment, and add the new token to it */
            TargetFrag = Translate (SourceFrag)
            Attach (TargetToken, TargetFrag)
            SourceInvSyntax = Parent (SourceFrag)
            TargetInvSyntax = Translate (SourceInvSyntax)
            Attach (TargetFrag, TargetInvSyntax)
            SourceFrag = Parent (SourceInvSyntax)
            /* Create new fragments and invocation syntax until TargetToken is
            connected to the tree. */
            WHILE (NOT IsOpen (Translation (SourceFrag)) ) BEGIN
                TargetFrag = Translate (SourceFrag)
                Attach (TargetInvSyntax, TargetFrag)
                SourceInvSyntax = Parent (SourceFrag)
                TargetInvSyntax = Translate (SourceInvSyntax)
                Attach (TargetFrag, TargetInvSyntax)
                SourceFrag = Parent (SourceInvSyntax)
            END WHILE
            TargetFrag = Translation (SourceFrag)
            Attach (TargetInvSyntax, TargetFrag);
        END ELSE
END FOR
```

Code Replication

While the target fragment tree usually has a shape similar to the source fragment tree, this method allows for code replication and reordering to produce new subtrees or subtrees in new places (when, for example, the replicated or reordered code represents an entire macro expansion).

The "for" loop from the example in the next example uses a macro invocation in place of the control variable.

pTAL Code	Generated C++ Code
define i(x) = an[x]#; for i(4) := 0 to 4 do	#define i(x) an[x] for (i(4) = 0; i(4) <= 4; i(4)++)

The Rosetta Translator preserves the semantics of the original code only if the code in the control variable of the for loop executes without side-effects. This issue is independent of the presence or absence of macro invocations in a for loop; the issue relates to the replication of code.

Experience with the Rosetta Translator

Automated translation is an important vehicle for migrating critical system code out of a proprietary programming language. Taken together, system products contain millions of lines of code, of which hundreds of thousands of lines are shared interfaces which contain many thousands of macros. These macros contain a wide variety of language constructs, and the language constructs are often not well-formed. The Rosetta Translator facilitates this migration. At this writing, preliminary translations have converted major parts of Tandem's core system software to C++.

This section describes the principles and requirements that guided the design of the Rosetta Translator.

Requirements of the Rosetta Translator

The Rosetta Translator was conceived as a way to reduce and eventually eliminate dependence on a proprietary systems programming language while supporting huge bodies of legacy software. To that end, the Rosetta Translator performs highly automatic translation from Tandem's portable Transaction Application Language (pTAL) to a subset of C++. The translated software is more portable and is compatible with commercially-available programming tools. New software can make use of the object-oriented programming paradigm while interoperating with legacy code. The resulting C++ source code is sufficiently readable and maintainable that the original pTAL source code can be retired.

The Rosetta Translator has several requirements:

The translation process must be highly automated. The Rosetta Translator must translate an entire product in a short elapsed time. The Rosetta Translator must be "highly automated," but not necessarily completely automated. The Rosetta Toolset includes a set of problem analysis tools to speed elapsed translation time. Automatic source-to-source translators can be criticized for automatically translating only a subset of the source language or for generating obfuscatory code, and source-to-source translators that decline to translate some features of a language can be criticized as useless when the source code is very large. The Rosetta Translator is an industrial strength toolset that must not substantially subset the source language for the convenience of the translator. It also must make use of commonly used idiomatic features of the target language, and must avoid translations that, while strictly safe, are not particularly readable or maintainable. The Rosetta Translator can take cues from a human expert to help it avoid generating awkward code.

The translation process must allow incremental translation on a module-by-module basis. To ease the transition from one implementation language to another, a product must be able to export translated interfaces before translating all of its implementation modules, and it must be able to phase translation of the product, one module at a time. The resultant C++ code must be interoperable with pTAL code.

The resultant code must faithfully reproduce the semantics of the original program.

The resultant code must be readable and maintainable. The generated program must be as readable and maintainable as the original code to a programmer familiar with both languages and the product that the code implements. A programmer familiar with the product must find that after translation, he or she still holds a valid mental model of the implementation. Identifiers must be mapped to reasonable spellings. Comments must appear adjacent to the code that they describe, even when that code has been moved. Patterns or pictures embedded in end-of-line comments must be preserved to the extent possible. The format of the code must appear natural in the target language, though some aspects of the format might follow the style and alignment of the original code, reflecting the preferences of the author. Macros defined in the source code must, to the extent possible, encapsulate the same meaning in the translated code.

The Rosetta Translator generates output that looks natural in its new language but does not significantly alter the design of the program. One of the great investments that software companies must make is in software maintenance, and one of the great assets that a software maintainer might have is familiarity with the implementation details of a large and complex software system. Substantial redesign is best done by the programmer, who is not only best qualified to make a complex judgement call, but can later make good use of the experience of doing so.

The resultant code must be efficient. Execution speed of a generated program must be comparable to that of the original pTAL-coded program. While fulfillment of this goal lies more in the hands of the compiler that implements the target language, we did have the opportunity to take into account performance of various constructs when designing their translations.

These requirements led to a number of challenges for the Rosetta Translator, including maintaining dual language interfaces, mapping tokens in language constructs, and dealing with inconsistently translated fragments.

Maintaining Dual Language Interfaces

Unfortunately, converting a product to C++ does not immediately obviate the need to use the pTAL language. Products that export a pTAL-coded interface must continue to export that interface in pTAL as well as C++ until no pTAL clients remain. Because of the size and complexity of externalized interfaces, and because of the opportunity for error inherent in manually maintaining parallel interfaces in both languages, it is essential that a product maintainer update just one copy of an interface and use a tool to generate the interface in the other language. The Rosetta Translator is that tool. The product maintainer uses the pTAL version as the reference copy.

This problem implies that the Rosetta Translator must emit code that is ready-to-roll. If any adjustment or enhancement to an interface file is necessary, then the adjustment or enhancement must be made to the pTAL code so the generated C++ can be read and used as is. Because pTAL does not support many useful C++ features, we supply a way to express new C++-only constructs in the pTAL source. The pTAL compiler ignores such constructs.

Objects in shared interfaces are only available to clients that have converted to C++ because the reference language, pTAL, cannot express them. Thus, a language conversion does not immediately clear the way for a paradigm shift visible in externalized interfaces.

Translating Language Constructs

Translation of each syntactic construct carries out a lexical mapping from source language lexical elements to target language lexical elements. The choice of target language construct and mapping of tokens from a source language construct to a target language construct is a matter of maximizing semantic correspondence between the lexical elements.

A few simple principles guide choices of construct translations in the Rosetta Translator. These principles are intended to make the resultant code appear more natural than most machine-generated code.

Generate simple resultant code to emulate simple source code, even if the transformation is not general. Generate complex resultant code only when necessary to emulate a difficult corner of the source language. For example, the pTAL language allows structure fields to overlay any other type field arbitrarily. If the space allocating field is at least as large as the fields that overlay it, the Rosetta Translator generates a C++ union. If it is not, the Rosetta Translator generates a macro invocation that, in turn, provides access to the appropriate memory location when the field is used. The next example depicts the translation of a structure type and variable.

pTAL Code	Generated C++ Code
struct s; begin int i; int j = i; int a; int(32) b = a; end;	class _s { public: union { short i; short j; }; short a; _redef (long, b, a); } s;

The Rosetta Translator generates references to macros, such as the invocation expression "_redef" in this example, that encapsulate the emulation of difficult pTAL features. These macros reside in an include file, "rosetta.h".

Use simplifying transformations when possible. For example, in C++ the address of the beginning of an array is easiest to express using the name of the array, so if the Rosetta Translator initially generates "&arrayname[0]", it applies a simplifying transformation to produce "arrayname."

Attempt to generate idioms commonly used in the target language. For example, C++ users frequently use the "++" idiom for incrementing variables; the Rosetta Translator generates this breed of idiom when appropriate.

When several equally plausible choices of coding style present themselves, preserve the style that the programmer used. For example, when choosing between C++ "*varname" and "varname[0]", the Rosetta Translator emulates the style of the pTAL code.

Strategies for Coping with Poorly Matched Language Constructs

Several strategies are possible to cope with translating one source language to another when the language constructs do not map precisely to one another at the token level. The Rosetta Translator uses the following strategies, each where it is most useful:

Restrict the source language to constructs that can be translated directly into the target language. This approach simplifies construct translation. The main disadvantage is that the user must either avoid using the untranslated constructs or correct the generated code. This is only practical when an equivalent supported construct exists, or when

virtually none of the legacy source code actually contains the forbidden construct. We chose to restrict certain source language constructs that cannot be expressed in the target language, such as taking the address of a label. We also chose to restrict certain source language features whose translation would especially lack readability, such as pTAL expressions that depend on order of evaluation. The restricted constructs can be recoded easily. The Rosetta Toolset offers semiautomatic help in removing dependencies on these features.

Extend the source program to give extra information that leads to a better translation. With a small amount of information from a programmer familiar with the source code, the Rosetta Translator can create much more natural looking generated code. The main disadvantage is that the user must modify his or her source code before translating.

The Rosetta Toolset offers tools to make that activity less tedious and less prone to error, and to make optimal use of a programmer's time. The Translator detects when a specific kind of information would be helpful and issues an appropriate diagnostic. A programmer familiar with the source code responds to the Rosetta Translator's request for semantic hints by embedding a special comment in the source code from which the Rosetta Translator can glean the extra information it needs. Supplying hints this way does not disturb the behavior of the source code and does not require extending the source language's compiler. The information is also retained in the original source code for future retranslations. The problem analysis tools simplify the tasks performed by the human expert by inserting the desired special comment in exactly the right place.

For example, the pTAL language has no declarations of unsigned variables, but instead offers unsigned operators that carry out unsigned arithmetic. A user can mark a pTAL variable as unsigned by inserting a particular comment into the variable declaration. The Rosetta Translator translates it as a true unsigned variable, obviating the need for type casts to force unsigned operations. The next example depicts the translation of an unsigned operation.

```
pTAL CodeGenerated C++ Code

int steve;short steve;

i :=steve '+'1234;i=(unsigned short)steve+1234;

!unsigned! int dano;unsigned short dano;

i :=dano '+'1234;i=dano+1234;
```

Hide the complexity of the translated code by emitting a meta-language (via macros) to implement very complex corners of the language. This is a portable way of extending the target language. This approach can preserve readability even when a simple construct from the source language translates to a fairly complex construct in the target language. The main disadvantage is that the software maintainer must learn how to use the special macros. For example, bit extraction and deposit operations on integers are supplied as macros in an include file, rosetta.h. The next example depicts the translation of bit extraction and bit deposit.

```
pTAL CodeGenerated C++ Code

x :=i.<4:5 >;x=_bit_extract(i, 4, 5);

i.<0:3>:=x;_bit_deposit(i, 0, 3, x);
```

Extend the target language to handle hardware dependencies that cannot be expressed in the standard target language. The target language was extended to support some special functional requirements of Tandem systems software. The target language was not extended merely to accommodate automatic translation. The Rosetta Translator translates systems programming constructs from the source language to target language extensions. For example, pragmas were added to produce similar data layout between C++ and pTAL.

Handling Comments

Source-to-source translators can use one of several strategies for translating comments that appear interspersed in the source language program:

Gather and translate comments separately from the source code, then collate the translated comments with the translated program text into the output stream [5]. This strategy simplifies translation by separating the issue of comment association from the problem of language construct translation. One disadvantage is that comments describing code that was reordered are difficult to place sensibly.

Associate comments with the syntactic construct in which they are embedded [1]. This strategy reorders comments and code together, and the comment placement issues do not clutter language construct translations. One disadvantage is that very fine-grain associations might not be preserved. Text-based preprocessors, such as document generators, that use special comments as cues might accept a new language as input more easily if finer-grain associations were preserved.

Associate comments with a nearby token. The Rosetta Translator's strategy is to bind comments closely to a token and to translate the comments associated with each token when the token is mapped to the target language. One disadvantage is that when source language tokens do not correspond one to one with target language tokens, the translations of syntactic constructs must explicitly specify comment mapping. This strategy trades off complexity for control, and increased control is desirable to preserve appropriate comment placement.

The Rosetta Translator is able to preserve end-of-line comments in their original form, which is especially useful when the information contained therein is tabular or pictorial in nature. All comments remain associated with the tokens they describe, even if part of the enclosing construct must be moved.

When a source token is mapped to a target token, comments tied to the former are automatically translated and attached to the latter. Routines implementing language construct translations must account for the comments associated with each source language token. If a source language token maps to no target language token, then its comments must be moved to an appropriate token that will be mapped.

The appearance of the generated code can benefit from construct translation routines that create comment associations that might differ from token associations. For example some construct translations appear more natural if the comments that appear before the first source language token in a particular construct also appear before the first target language token in the translated construct, even if the first source token does not map to the first target token.

Comments scanned in fragments that contain no tokens are associated with the static fragment, translated separately from the tree translations, and appear in the analogous C++ static fragment.

Fragment Consistency

Because the Rosetta Translator's design requires that each source language macro maps to exactly one target language

macro, it checks that every macro use is textually identical, to ensure that the macro text "works" in all contexts of use. It also checks included files. The scope of a macro actual parameter is limited to the macro body itself, so the Rosetta Translator need only check that all expansions of a formal parameter within a given macro expansion are textually identical.

This and the next few sections discuss diagnosing fragment textual mismatches, strategies for minimizing mismatches, and remedies for mismatches that cannot be prevented.

Fragments Containing Puns

When the syntax for expressing two different operations is different in the target language but the same in the source language, a fragment translation might not "work" in every context of use. In programming languages as in natural language, puns tend to translate poorly. When type compatibility rules differ between the two languages, for example, type casts can lead to a mismatch in macros that are used in conjunction with multiple types.

The following example illustrates an error condition: the translation of two expansions of the same macro result in different macro body text. All instances of a macro body must be checked for consistency before output source generation. The Rosetta Translator generates references to macros, such as "_redef" and "_tobitfield" in this example, that encapsulate the emulation of difficult pTAL features. In this case, the integer field "a" overlays the bit field "k". The "_tobitfield" macro takes the name of the surrounding class type as a parameter. That name is different for the two uses of the macro: "f_" versus "g_". The next example depicts a macro whose two uses result in macro body text that does not match; mismatching tokens are in bold.

pTAL Macro	Generated C++ Macro
!zero-length array causes a to overlay k! define fields = *ERROR* unsigned(1) k #;	int a [0:-1]; #define fields

pTAL Code	Generated C++ Code
struct f; begin fields; end; struct g; begin fields; end;	class f_ { fields; } f; class g_ { fields; } g;
pTAL Code Expansion	C++ Code Expansion
struct f; begin k)); int a [0:-1]; unsigned(1) k; end; struct g; begin k)); int a[0:-1]; unsigned(1) k; end;	class f_ { _redef(short, a, _tobitfield(f_, unsigned k:1; } f; class g_ { _redef(short, a, _tobitfield(g_, unsigned k:1; } g;

Blurred Fragment Boundaries

If a construct translation blurs fragment boundaries, the translated fragment tree looks as if two instances of the fragment were created, and the contents of those fragments are not textually identical. Fragment boundary violations are detected by the same fragment consistency check that detects puns that do not work. The following example illustrates a blurred fragment boundary in a use of an ill-formed macro.

If precedence rules differ between the two languages, then introducing parentheses to preserve the original expression's semantics can lead to a consistency violation. The pTAL language assigns higher precedence to the left shift operator than to the addition operator. The C++ language assigns higher precedence to the addition operator than to the left shift operator. Parentheses appear in the target language version of the expression to retain the order of evaluation. The parentheses tokens happen to be generated from the higher precedence operator, which determines their fragment assignment.

pTAL Code	Expansion of pTAL Code	Generated C++ Code
define fred(p) = p << 2#; ... a := fred (x + 1) ;	a := x + 1 << 2;	a = x + (1 << 2) ;

The translation of the previous example results in a C++ fragment tree that appears to have two instances of the actual parameter fragment which do not contain identical text.

Strategies for Minimising Inconsistently Translated Macros and Macro Actual Parameters

Inconsistently translated macro bodies and macro actual parameters can be minimized by building special rules into the token mapping methods, when possible.

Suppress simplifying transformations on the generated code when inside of a macro expansion. Generally, syntax transformations are not sensitive to whether or not the generated code is inside of a macro body expansion or macro formal parameter expansion. Some routines that perform construct translations are sensitive to fragment boundaries, though they do not translate fragments. These routines suppress simplifying transformations within a macro expansion or macro formal parameter expansion in order to make translations for different macro invocations textually identical. For example, the transformation simplifying the text "&arrayvar[0]" to "arrayvar" is suppressed within a macro when the index is in a macro parameter. Likewise, the transformation simplifying the text "(*ptr).field" to "p->field" is suppressed when "*p" is a macro actual parameter.

Absorb tokens that differ between invocations of a macro into an actual parameter, when possible. Source language tokens might generate not only the logical translation of themselves, but also some attendant baggage. If that baggage can differ in two different invocations of the same macro, there is no problem if the baggage is mapped to the actual parameter rather than the macro body. For example, C++ parentheses generated to force the appropriate operator precedence should be absorbed into the translated actual parameter if they surround C++ tokens generated only from pTAL actual parameter tokens.

Likewise, if a C++ "*" operator was generated in the translation of an implicitly dereferenced pointer in a macro actual parameter, the "*" operator should be absorbed into the translated actual parameter so that nonpointer variables can be passed into the macro.

Extrude tokens that differ between invocations of a macro into the fragment containing the macro invocation syntax.

Similar to the strategy previously described, if the differing tokens are on the edge of the C++ translation of a macro body, there is no problem if they are mapped outside. For example, if a C++ type cast applies to an entire macro expansion, then the generated text should show the type cast applied to the macro invocation syntax.

Remidies for Inconsistently Translated Macros and Macro Actual Parameters

Although a careful design of token associations, taking into account common coding practices in the source language, can ensure that fragment mismatch problems are a rare event, it cannot prevent all possible problems. When an inconsistently translated macro or macro actual parameter occurs, the user can choose from three remedies. Each of the remedies can solve any fragment mismatch problem. If the macro in question is exported, changing the macro interface in the original language implementation is usually not possible.

Recode the macro body so that the translation "works." This strategy is viable if recoding the body of the macro avoids the problematic construct and does not disturb the interface or the usage of the macro. If recoding the macro requires changes at all points of use, then this might not be a viable solution.

Give the Rosetta Translator permission to expand all uses of the macro. This strategy eliminates any fragment mismatch problem at the cost of deimplementing the macro that contained a problem. It is not a desirable solution if the macro was useful for improving the usability, maintainability, or readability of the software, or if the macro was exported. It is a viable solution if the product maintainer is not interested in preserving the macro. The next example illustrates the expansion of a macro.

pTAL Code	Generated C++ Code
define !expand! shifty (x, y) = x << y#; a := shifty(2, 3);	a = 2 << 3;

Give the Rosetta Translator permission to create an additional macro parameter to encapsulate the differences. This strategy eliminates any fragment mismatch problem. It has the potential of producing actual parameters in the target language code that are less sensible than a programmer would write. Recalling the example of a mismatching macro body from the previous section, the differences can be encapsulated. The next example illustrates this encapsulation.

pTAL MAcro	Generaed C++ Macro
!zero-length array causes a to overlay k! define fields !FORMAL:stype! = int a [0:-1]; unsigned(1) k #; _tobitfield(stype, k)); \	#define fields(stype) \ redef(short, a, unsigned k: 1

pTAL Code	GeneratedC++ Code
struct f; begin fields; end; struct g;	class f_ { fields (f_); } f; class g_ {

-continued

begin fields; end;	fields(g_); } g;

References

1. Albrecht, P. F., Garrison, P. E., Graham, S. L., Hyerle, R. H., Ip, P., Krieg-Bruckner, B., "Source-to-Source Translation: Ada to Pascal and Pascal to Ada". *Proceedings of the ACM SIGPLAN Symposium on the Ada Programming Language*, November, 1980.

2. Atkinson, R., Demers, A., Hauser, C., Jacobi, C., Kessler, P., and Weiser, M., "Experiences Creating a Portable Cedar". *Proceedings of the SIGPLAN '89 Conference on Programming Language Design and Implementation*, June, 1989.

3. Bartlett, J., *SCHEME->C, a Portable Scheme-to-C Compiler.* Technical Report, Digital Western Research Laboratory, 1989.

4. Lemkin, P., "PSAIL: SAIL to C", *Computer Language*, 2(8), August, 1985.

5. Moynihan, V. D., and Wallis, P. J. L., "The Design and Implementation of a High-Level Language Converter", *Software—Practice and Experience*, 21(4), April, 1991.

6. Stroustrup, B., *The C++ Programming Language*, Addison-Wesley, 1991.

7. Tandem Computers, Inc., *Transaction Application Language (TAL) Reference Manual*, Version D10.

8. Waters, R. C., "Program Translation via Abstraction and Reimplementation", *IEEE Transactions on Software Engineering*, 14(8), August, 1988.

9. Weiner, J., Ramakrishnan, S., "A Piggy-back Compiler For Prolog", *Proceedings of the SIGPLAN '88 Conference on Programming Language Design and Implementation*, June, 1988.

DESCRIPTION OF THE PREFERRED EMBODIMENT

A prior art tree data structure is outlined in FIG. 1. The purpose of this data structure is to relate a set of "things," or objects in a hierarchy. One object in this set, called a root node, is at the vertex. From the root, links, called branches, connect to other objects, called nodes. In turn, the nodes may branch to other nodes ultimately terminating in a set of objects called leaves. Viewing the tree from the root to the leaves, tracing a single path of branches, nodes are ancestors of each other. Conversely, viewing the tree from a leaf to the root, again tracing a single path, nodes are descendants of each other. Choosing an arbitrary node, it is considered the child of the node to which it branches toward the root and the parent of the node branching toward the leaves.

FIG. 2 illustrates a contribution of the present invention to the prior art. It is termed a doubly-rooted tree data structure. This doubly-rooted tree has two roots **511** and **517**, but only a single set of leaves **515**. From each root emanates a tree, as in the prior art, but the nodes **513** and **519**, branches **521** and **523**, and hierarchical arrangement of the two trees may be different as long as the two trees co-terminate at a single set of leaves **515**. In the present invention, the leaves of a doubly-rooted tree are considered tokens and are collectively referred to as the "virtual source."

FIG. 2A is a simplified block diagram of a computer system, designated generally with the reference numeral

10A incorporating the data structure of FIG. 2. As FIG. 2A illustrates, the computing system 10A includes a central processing unit (CPU) 11A that is coupled to a memory 14A and an input/output channel (IOC) 16A by a bus structure 18A. The IOC 16A, in turn, connects the CPU 11A to a data storage area 22A by an input/output (I/O) 20A. The data storage area 22A may take the form of a magnetic disk media storage device, an electronic memory, or a communications link to other storage areas.

While a doubly-rooted tree is two dimensional, the concept can be expanded to three, four, or more dimensions (not illustrated). Multiple trees may co-terminate at a single set of leaves to form an N-rooted tree.

Prior art trees are useful in computer language compilers such as the simple one depicted in the data flow chart, FIG. 3. In the prior art compiler, source code 51 flows first through a scanner 53, where a stream of characters is divided into recognizable "words" or tokens and then to a parser 55, which determines a hierarchical relationship among the tokens and builds a syntax tree 57. An additional step, performed by a semantic analyzer 59, further refines the syntax tree 57. A syntax translator 61 converts the syntax tree 57 into an intermediate code tree 63. In turn a code generator 65 converts the intermediate code tree 63 into object code 67. Also commonly called machine code, object code 67 is an instruction sequence which a computer can directly execute and is usually stored on magnetic media or in electronic memory. In the normal course of usage, machine code is not stored in a manner which humans can read and interpret. For background information on the construction of compilers, see Aho, Sethi, and Ullman, *Compilers: Principles, Techniques, and Tools*, Addison-Wesley, 1985.

FIG. 3A is a close view of the syntax tree 57 for the prior art compiler depicted in FIG. 3. It represents the FORTRAN statement *Z=X+Y* (nodes *57a, 57b, 57c, 57d, 57e*).

As illustrated in FIG. 4, the source code to source code translator of the present invention, in simplified form, generally follows the flow of the prior art compiler of FIG. 3. However, instead of employing simple tree data structures, the translator uses two doubly-rooted trees to preserve contextual information such as macros, conditionally compiled regions of code, source inclusion, comments, and other preprocessor characteristics.

FIG. 4A is a close view of the source syntax tree 7 and the joined leaves 9, representing virtual source, for the present translator depicted in FIG. 4. It represents the FORTRAN statement *Z=X+Y* (nodes *7a, 7b, 7c, 7d, 7e* and leaves *9a, 9b, 9c, 9d, 9e*). The leaves 9 representing virtual source are absent in the syntax tree 57 for the prior art compiler depicted in FIG. 3A. The prior art compiler does not require a complete virtual source to reduce source code to intermediate language or machine code, while the present invention does require virtual source to translate from one high level language to another.

FIG. 5 is a flowchart that describes a method for building of the first tree in doubly rooted tree, joined at the leaves. This method is applicable to building the source fragment tree 19 of FIG. 4. The method builds the tree from the top down. The flowchart uses the following data and operations:

Source is a set of ordered set of objects employed in step 603, one of which is identified as primary.

Ordered set of objects, operations:

Open 603

Test for whether or not the set is exhausted 605

Get the next object 607

Object is a leaf or an ordered set

Operations:

Test for whether the object is a leaf or an ordered set 609

Attach to parent node

Node operations:

Accept ordered attachment of children

Get parent node

FIG. 6 is a flowchart that describes a method for building a second tree in the doubly rooted tree, joined at the leaves. This method is applicable to building the source syntax tree 7 of FIG. 4. The ordered set of leaves in step 621 that is input to the parser 623 is yielded by an inorder traversal of the first tree. The method builds the tree from the bottom up.

FIG. 7 is a flowchart that describes the translation of the second tree in the doubly rooted tree, joined at the leaves.

FIG. 8 is a flowchart that describes the translation of the first tree in the doubly rooted tree, joined at the leaves. The flowchart uses the following data and operations:

Node operations:

Translate

Attach to parent node

Test for whether the node is "open": The "open" nodes in the tree include the ancestors of the last leaf that was added to the tree. Adding a child to a node "closes" nodes that are no longer ancestors of the latest leaf added.

Leaf operations:

Attach to parent node

Input set operations:

Test for whether the set is exhausted

Another way of describing FIG. 8 is that it shows a method of translating a fragment tree from the source doubly-rooted tree to the target doubly-rooted tree. This is a critical stage in the translation process; it is the point at which preprocessor characteristics are transferred from the source code to the target code. After building the target language syntax tree, the translator traverses the target language virtual source token stream, and builds the target language fragment tree, based on the position in the source language fragment tree of the source language token associated with each target language token encountered in the virtual source stream.

Operations are as follows:

Attach(x, y): Attach x to y: y becomes the parent of x. x is added positionally after pre-existing siblings.

Translate(n): Translate n (token or fragment) to the target language version of the object, and return the translation.

Translation(n): Return the latest translation of the fragment n; nil if it has not been translated.

Parent(n) : If n is a token, return the fragment in which n resides. If n is a fragment, return the invocation syntax of the fragment (macro name token, for example).

IsOpen(n): Return true if the fragment n is open in the target fragment tree. The "open" nodes in the target language fragment tree include the ancestors of the last token that was added to the fragment tree. Adding a child to a fragment "closes" nodes that are no longer ancestors of the latest token added. Begin by translating the main source include fragment. This fragment is initially open.

```
FOR (TargetToken = each token in the target language virtual
soucre) BEGIN
    SourceToken = the source token associated with the current
TargetToken
    SourceFrag = Parent (SourceToken)
```

-continued

```
    IF (IsOpen (Translation (SourceFrag)))
    THE BEGIN
        /* add a new token to an existing fragment that is already
open */
        TargetFrag = Translation (Source Frag)
        Attach (TargetToken, TargetFrag)
    END ELSE BEGIN
        /* create a new fragment, and add the new token to it */
        TargetFrag = Translate (SourceFrag)
        Attach (TargetToken, TargetFrag)
        SoucreInvSyntax = Parent (SourceFrag)
        TargetInvSyntax = Translate (SourceInvSyntax)
        Attach (TargetFrag, TargetInvSyntax)
        SourceFrag = Parent (SourceInvSyntax)
        /* Create new fragments and invocation syntax until
    TargetToken is
            connected to the tree. */
        WHLE (NOT IsOpen (Translation (SourceFrag)) ) BEGIN
            TargetFrag = Translate (SoucreFrag)
            Attach (TaretInvSyntax, TargetFrag)
            SourceInvSyntax = Parent (SourceFrag)
            TargetInvSyntax = Translate (SourceInvSyntax)
            Attach (TargetFrag, TargetInvSyntax)
            SourceFrag = Parent (SourceInvSyntax)
        END WHILE
        TargetFrag = Translation (SourceFrag)
        Attach (TargetInvSyntax, TargetFrag);
    END ELSE
END FOR
```

Taken together, FIGS. **9**, **10**, and **11** show an example of how the method of FIGS. **5**, **6**, **7**, and **8** operates on a specific two line macro:

```
define less(x)=a<x#;
while less(42) do
```

The macro was written for a compiler of a Tandem proprietary high level language, pTAL. The target source code is a well known standardized language, C++. Correctly translated to C++, the macro reads:

```
#define less(x)<x
while (a less(42))
```

FIG. **9** depicts an example fragment tree and syntax tree, using the pTAL language as the source language. This diagram represents an explosion of the pTAL syntax tree, pTAL fragment tree, and the pTAL tokens that connect them depicted in a more general way in FIG. **4**: the source fragment tree **19**, the source syntax tree **7**, and leaves **9** that join them. The pTAL fragment tree and syntax tree is a doubly rooted tree jointed at the tokens. The tokens making up the body of the define "less" are represented in their context of use.

Fragment **121** represents the primary input file. Fragment **127** represents the invocation of the macro "less". Tokens **153**, **155**, **157**, **159**, **161**, **163**, **165**, **167**, **169**, **129**, **131**, **135**, and **177** are in fragment **127**. The macro's invocation syntax **129**, **131**, **135** is associated with the fragment **127**. The tokens **171**, **173**, and **141**, which compose the macro body, are in fragment **127**. Fragment **137** represents the substitution for macro formal parameter "x" of the actual parameter text, which is the token **175**. The macro formal parameter's invocation syntax **141** is associated with the fragment **137**. The token **175**, which is the macro actual parameter, is in fragment **137**.

Node **179** represents a pTAL procedure. That procedure is composed of a macro declaration represented by node **147** and a while statement represented by node **149**. The macro declaration **147** contains the tokens **153**, **155**, **157**, **159**, **161**, **163**, **165**, and **167**. The body of the macro is expanded in its context of use. The while statement **149** is composed of the token **169**, an expression represented by node **151**, and the token **177**. The expression **151** is composed of the tokens **171**, **173**, and **175**.

FIG. **10** depicts a pTAL fragment subtree and syntax subtree, including the tokens that link them. It also depicts the C++ syntax tree that represents the translation of the pTAL syntax tree. The syntax translator has translated each pTAL syntactic construct to a semantically equivalent C++ syntactic construct. In addition, it maps pTAL tokens to C++ tokens. The pTAL fragment tree has not yet been translated to its C++ counterpart. The source language while statement **149** has been translated to the target language while statement **239**. The source language expression **151** has been translated to the target language expression **241**. Each C++ token in the C++ syntax tree **243**, **245**, **247**, **249**, **251**, **253** was "generated" by exactly one pTAL token. C++ token **243** was generated by pTAL token **169**. C++ token **245** was generated by pTAL token **171**, and so on as depicted by dashed arrows. The association between a C++ token and a pTAL token, depicted as dashed arrows, determines in which virtual fragment the token belongs; these associations make it possible to build a target language fragment tree. One pTAL token may generate several C++ tokens: **169** generates **243**, **245**, and **253**. One pTAL token may generate no C++ tokens: token **177** generates no C++ tokens. Each C++ token is generated by exactly one pTAL token.

FIG. **11** depicts a pTAL fragment tree and syntax tree, including the tokens that link them. It also depicts a semantically equivalent C++ fragment tree and syntax tree, including the tokens that link them. It depicts FIG. **11** after the translation of the fragment tree. In order to perform the translation, traverse the leaves of the C++ syntax tree; these leaves will become the leaves of the C++ fragment tree.

Following is a walkthrough of building the example C++ fragment subtree that is equivalent to the pTAL fragment subtree depicted in FIG. **10** and FIG. **11**. A fragment in the C++ fragment tree is "open" if it is an ancestor of the leaf that was most recently added to the C++ fragment tree.

Translate the root of the pTAL fragment tree: the pTAL fragment **121** is translated to the C++ fragment **273**. The C++ fragment tree root node **273** is initially "open".

Consider the C++ token **243**.

This token is associated with the pTAL token **169**.

The pTAL token **169** is contained in pTAL fragment **121**, which translated to C++ fragment **273**.

C++ fragment **273** is "open".

Add C++ token **243** to C++ fragment **273**.

Consider the C++ token **245**.

This token is associated with the pTAL token **169**.

The pTAL token **169** is contained in pTAL fragment **121**.

pTAL fragment **121** translated to C++ fragment **273**.

C++ fragment **273** is "open".

Add C++ token **245** to C++ fragment **273**.

Consider the C++ token **247**.

This token is associated with the pTAL token **171**.

The pTAL token **171** is contained in pTAL fragment **127**.

The pTAL fragment **127** has not been translated, so does not appear in the C++ fragment tree.

Translate the pTAL fragment **127** to create the C++ fragment **265**.

Add the C++ token **247** to the C++ fragment **265**.

The parent of the pTAL fragment **127** is the pTAL fragment **121**.

pTAL fragment **121** translated to the C++ fragment **273**.

C++ fragment **273** is "open" in the C++ fragment tree.

Add the C++ fragment **265** to the C++ fragment **273**.

Consider the C++ token **249**.

This token is associated with the pTAL token **173**.

The pTAL token **173** is contained in the pTAL fragment **127**.

The pTAL fragment **127** translated to the C++ fragment **265**.

The C++ fragment **265** is "open" in the C++ fragment tree.

Add the C++ token **249** to the C++ fragment **265**. Consider the C++ token **251**.

This token is associated with the pTAL token **175**.

The pTAL token **175** is contained in the pTAL fragment **137**.

The pTAL fragment **137** has not been translated, so does not appear in the C++ fragment tree.

Translate the pTAL fragment **137** to create the C++ fragment **261**.

Add the C++ token **231** to the C++ fragment **261**.

The parent of the pTAL fragment **137** is the pTAL fragment **127**.

The pTAL fragment **127** was translated to the C++ fragment **265**.

The C++ fragment **265** is "open" in the C++ fragment tree.

Add the C++ fragment **261** to the C++ fragment **265**. Consider the C++ token **253**.

This token is associated with the pTAL token **169**.

The pTAL token **169** is contained in pTAL fragment **121**, which translated to C++ fragment **273**.

C++ fragment **273** is "open".

Add C++ token **253** to C++ fragment **273**.

Notice that the C++ fragments **261** and **265** are no longer "open": they are no longer ancestors of the last C++ token to be added to the C++ fragment tree.

What is claimed is:

1. A method for use in a computer system for translating a first source code of a source computer language into a second source code of a target computer language, said method comprising:

creating a source doubly-rooted tree data structure, said creating including:

retrieving the first source code;

producing virtual source from the first source code to create a source fragment tree;

scanning the first source code to form a virtual source;

identifying the virtual source as leaves of the source doubly-rooted tree;

identifying leaves of the source fragment tree with the leaves of the source doubly-rooted tree;

parsing the virtual source into a source syntax tree such that the leaves of the source syntax tree are identified with the leaves of the source doubly-rooted tree; and

obtaining and storing semantic information completing a description of the source code in the source syntax tree;

creating a target doubly-rooted tree data structure including:

translating the source syntax tree and creating a target syntax tree;

identifying leaves of the target syntax tree with leaves of the target doubly-rooted tree;

associating leaves of the target doubly-rooted tree with corresponding leaves of the source doubly-rooted tree;

translating the source fragment tree and creating a target fragment tree so that leaves of the target fragment tree are ordered; and

identifying leaves of the target fragment tree with leaves of the target doubly-rooted tree;

generating the second source code from the target doubly-rooted tree; and

storing the second source code.

2. The method of claim **1** wherein the step of translating the source fragment tree and creating a target fragment tree comprises:

locating the root of the source fragment tree; translating the source root to the root of the target fragment tree; and

building the target fragment tree based on the position of leaves in the source fragment tree including the steps of:

locating a target fragment tree leaf;

examining a source fragment tree leaf associated with the target fragment tree leaf;

examining a parent node of the source fragment tree leaf;

firstly testing whether the most recent translation of the parent node is an ancestor of the most recently attached target fragment tree leaf or is the root of the target fragment tree;

attaching the target fragment tree leaf to the parent node of the most recently attached leaf; attaching the target fragment tree leaf to the parent node; and examining a next target fragment tree leaf, if any, if the first test result is TRUE;

translating the parent node; adding the parent node to the target fragment tree; attaching the target fragment tree leaf to the parent node; assigning the parent node to a first temporary object;

assigning a parent of the parent node to a second temporary object . if the first . test result is FALSE;

secondly testing whether the second temporary object is an ancestor of the most recently attached target fragment tree leaf or is the root of the target fragment tree, if the first test result is FALSE;

adding the second temporary object to the target fragment tree; attaching the first temporary object to the ancestor node; examining a next target fragment tree leaf, if any, if the first test result is FALSE and the second test result is TRUE; and

translating the second temporary object; attaching the second temporary object to the fragment tree; attaching the first temporary object to the second temporary object; assigning the second temporary object to the first temporary object; and continuing from said step of secondly testing, if the first test result is FALSE and the second test result is FALSE.

3. An apparatus for translating a first source code of a source computer language into a second source code of a target computer language while preserving text preprocessor statements and expressions comprising:

source computer language file means with memory allocated for storing the first source code;

target computer language file means with memory allocated for storing the second source code;

means for creating a first and second doubly-rooted tree data structures with memory allocated for information to be stored within the first and second data structures;

means for retrieving the first source code from memory;

means for scanning and parsing the first source code into a first virtual source;

means coupled to the scanning and parsing means for storing the first virtual source code while scanning and parsing in the first doubly-rooted tree data structure so that text preprocessor statements and expressions are preserved;

means for translating the first doubly-rooted tree structure into a second virtual source code;

means coupled to the translating means for storing the second virtual source code in the second doubly-rooted tree data structure so that text preprocessor statements and expressions are preserved;

means for traversing the second data structure coupled to means for creating the second source code;

means for storing the second source code coupled to the target computer language file means.

* * * * *

APPENDIX C

CITED AUTHORITIES

United States Court of Appeals,
Federal Circuit.

ECOLOCHEM, INC., Plaintiff-Appellant,
v.
SOUTHERN CALIFORNIA EDISON COMPANY,
Defendant-Appellee.

No. 99-1043.

Sept. 7, 2000.
Rehearing Denied; Suggestion for Rehearing En Banc Declined Dec. 13, 2000.

Patentee of deoxygenation processes for liquid containing dissolved oxygen brought patent infringement action against operator of nuclear generating station. Operator counterclaimed for declaratory judgment of invalidity, and asserted equitable defenses. The United States District Court for the Central District of California, 863 F.Supp. 1165, granted partial summary judgment for operator, invalidating certain claims. The Court of Appeals affirmed in part and reversed in part. Following bench trial, the District Court, Richard A. Paez, J., 1998 WL 1182000, found willful infringement, but determined that all infringed claims were invalid. Patentee appealed. The Court of Appeals, Michel, Circuit Judge, held that: (1) patent claims were not anticipated by prior art articles; (2) one claim was anticipated by scientist's public presentation; (3) majority of claims were not rendered obvious by prior art, absent motivation to combine prior art references; but (4) patentee did not rebut prima facie case of obviousness as to one claim.

Affirmed in part, reversed in part, and remanded.

West Headnotes

[1] Patents ⚖70
291k70 Most Cited Cases

Claims in patent for deoxygenation processes for liquid containing dissolved oxygen were not anticipated by prior art articles, since articles' discussion of applications of deoxygenated water was limited to use of hydrogen deoxygenation, while patent claims involved deoxygenation by hydrazine in combination with a mixed bed ion exchange resin. 35 U.S.C.A. § 102(a).

[2] Patents ⚖324.55(4)
291k324.55(4) Most Cited Cases

Court of Appeals reviews for clear error the district court's decision on anticipation of a patent claim after trial. 35 U.S.C.A. § 102(a).

[3] Patents ⚖60
291k60 Most Cited Cases

Claims in patent for deoxygenation processes for liquid containing dissolved oxygen, which required removal of excess hydrazine in the final ion exchange step, were not anticipated by scientist's public presentation at International Water Conference, which did not discuss removal of excess hydrazine, but presentation did anticipate claim directed to deoxygenation process comprising steps of passing liquid containing oxygen and hydrazine through activated carbon, and then passing liquid through mixed bed ion exchange resins to remove at least the dissolved contaminants. 35 U.S.C.A. § 102(a).

[4] Patents ⚖60
291k60 Most Cited Cases

A public presentation indicative of the state of knowledge and use of an invention in the United States qualifies as prior art for anticipation purposes under the patent statute. 35 U.S.C.A. § 102(a).

[5] Patents ⚖324.55(4)
291k324.55(4) Most Cited Cases

Court of Appeals reviews de novo the district court's conclusion of obviousness of a patent claim. 35 U.S.C.A. § 103(a).

[6] Patents ⚖16.5(1)
291k16.5(1) Most Cited Cases

Patent claims directed to deoxygenation processes for liquid containing dissolved oxygen, which used process set forth in prior art article, including use of carbon bed, followed by use of mixed bed ion exchange resin downstream of carbon bed, were not rendered obvious by combination of article and other references, as there was no evidence of any suggestion, teaching, or motivation to combine article and other references, and there was evidence of teaching away from demineralization and deoxygenation processes used by patentee. 35 U.S.C.A. § 103(a).

[7] Patents ⚖16(4)
291k16(4) Most Cited Cases

Court of Appeals cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention pursuant to a claim that patent is invalid for obviousness. 35 U.S.C.A. § 103(a).

[8] Patents ⚖26(1)
291k26(1) Most Cited Cases

When a rejection for obviousness depends on a combination of prior art references, there must be some teaching,

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

suggestion, or motivation to combine the references, and the same principle applies to invalidation of the patent. 35 U.S.C.A. § 103(a).

[9] Patents ⚷16(4)
291k16(4) Most Cited Cases

[9] Patents ⚷26(1)
291k26(1) Most Cited Cases

Although the suggestion to combine prior art references may flow from the nature of the problem, defining the problem in terms of its solution reveals improper hindsight in the selection of the prior art relevant to obviousness; therefore, when determining the patentability of a claimed invention which combines two known elements, the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination. 35 U.S.C.A. § 103(a).

[10] Patents ⚷36(1)
291k36(1) Most Cited Cases

Broad conclusory statements regarding the teaching of multiple prior art references, standing alone, are not evidence of a motivation to combine those references, as would support a claim of obviousness. 35 U.S.C.A. § 103(a).

[11] Patents ⚷36.1(2)
291k36.1(2) Most Cited Cases

[11] Patents ⚷36.1(3)
291k36.1(3) Most Cited Cases

[11] Patents ⚷36.2(1)
291k36.2(1) Most Cited Cases

Patentee failed to rebut prima facie case of obviousness as to claim for deoxygenation process for liquid containing dissolved oxygen, despite some evidence of secondary considerations of nonobviousness such as modest commercial success and teaching away from use of prior art process incorporated in claimed process, in light of absence of any evidence that others were trying to emulate patented process, and fact that patentee applied for patent on the process within two years of publication of industry guidelines recommending use of deoxygenated water in nuclear power plants. 35 U.S.C.A. § 103(a).

[12] Patents ⚷324.5
291k324.5 Most Cited Cases

[12] Patents ⚷324.55(4)
291k324.55(4) Most Cited Cases

While Court of Appeals review the district court's factual findings on the secondary considerations of obviousness for clear error, on claim of patent invalidity, Court reviews the ultimate determination of obviousness de novo. 35 U.S.C.A. § 103(a).

[13] Patents ⚷36.2(4)
291k36.2(4) Most Cited Cases

[13] Patents ⚷324.55(4)
291k324.55(4) Most Cited Cases

District court clearly erred in finding that operator of nuclear generating station established that success of patentee's invention, namely, a deoxygenation process for liquid containing dissolved oxygen, was due to factors not claimed in the patented invention, for purpose of claim that patent claim was invalid as obvious, in light of operator's own statements that success was based both on mobility of commercial embodiment of invention, which was not claimed, and on invention's improved filtration process, which was claimed. 35 U.S.C.A. § 103(a).

[14] Federal Courts ⚷844
170Bk844 Most Cited Cases
Court of Appeals gives great deference to the district court's decisions regarding credibility of witnesses.

[15] Patents ⚷34
291k34 Most Cited Cases

The fact of near-simultaneous invention, though not determinative of statutory obviousness, is strong evidence of what constitutes the level of ordinary skill in the art, although the possibility of near simultaneous invention by two or more equally talented inventors working independently may or may not be an indication of obviousness when considered in light of all the circumstances. 35 U.S.C.A. § 103(a).

[16] Patents ⚷36(3)
291k36(3) Most Cited Cases

District court clearly erred in finding that prior art did not teach away from using process set forth in prior art article, or from combining it with a mixed bed ion exchange resin in any manner, to develop claimed deoxygenation and demineralization process used in nuclear power generator, for purpose of claim that patent directed to deoxygenation process was obvious, in view of various references warning against use of prior art process due to carbon contaminants, and reference finding prior art process to be inefficient and expensive. 35 U.S.C.A. § 103(a).

[17] Patents ⚷36.1(2)
291k36.1(2) Most Cited Cases

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

[17] Patents ⚖36.1(5)
291k36.1(5) Most Cited Cases

One indicia of non-obviousness of a patented product is the acclamations it receives when it is released, and the copying that occurs. 35 U.S.C.A. § 103(a).

[18] Patents ⚖36.1(2)
291k36.1(2) Most Cited Cases

Court's belief that patented process was not novel was not proper basis for discounting copying of claimed invention as evidence of nonobviousness. 35 U.S.C.A. § 103(a).

[19] Patents ⚖36.1(2)
291k36.1(2) Most Cited Cases

A showing of copying of a claimed invention is only equivocal evidence of nonobviousness in the absence of more compelling objective indicia of other secondary considerations, because the alleged copying could have occurred out of a general lack of concern for patent property. 35 U.S.C.A. § 103(a).

Patents ⚖328(2)
291k328(2) Most Cited Cases

4,430,226. Cited As Prior Art.

Patents ⚖328(2)
291k328(2) Most Cited Cases

4,556,492. Cited.

Patents ⚖328(2)
291k328(2) Most Cited Cases
4,818,411. Invalid.

***1363** Clifton E. McCann, Lane, Aitken & McCann, of Washington, DC, argued for plaintiff-appellant. With him on the brief was Andrew C. Aitken.

Ted G. Dane, Munger, Tolles & Olson, of Los Angeles, California, argued for defendant-appellee. With him on the brief was Gregory P. Stone.

Before MICHEL, CLEVENGER, and RADER, Circuit Judges.

MICHEL, Circuit Judge.

Ecolochem, Inc. ("Ecolochem") filed suit in 1992, alleging that Southern California Edison Company ("Edison") infringed Ecolochem's U.S. Patent Nos. 4,556,492 ("the '492 patent") and 4,818,411 ("the '411 patent") when deoxygenating water in the High-Flow Makeup Demineralizer ("HFMUD") at Edison's San Onofre Nuclear Generating Station ("SONGS"). Edison denied infringement, counterclaimed for declaratory judgment of invalidity, and asserted equitable defenses. By grant of partial summary judgment to Edison, the United States District Court for the Central District of California invalidated claims 1, 2, and 5-10 of the '492 patent and claims 20-21 of the '411 patent, holding the subject matter of each of these claims to be either anticipated under 35 U.S.C. § 102 and/or obvious under 35 U.S.C. § 103. On appeal to this court, we reversed the holding by the district court that there was no genuine issue of material ***1364** fact that the invention of claim 20 of the '411 patent would have been obvious at the time of the invention, and remanded the case for a trial on invalidity in light of Ecolochem's evidence of secondary considerations. As to the invalidation of the other appealed claims, we affirmed. Ecolochem continued to assert infringement of claims 1, 3-13, 15, 17, 18, and 20 of the '411 patent after remand, but dropped its suit as to the remaining claims of the '492 patent. After a bench trial, the district court found that Edison had willfully infringed claims 1, 3-13, 15, 17, 18, and 20 of Ecolochem's '411 patent and rejected Edison's equitable defenses. The court then went on to invalidate all of the claims found to be infringed. Ecolochem appeals the holdings of invalidity as to claims 1, 3-13, 15, 17, 18, and 20 of the '411 patent here. Edison does not cross-appeal the district court's finding that Edison willfully infringed those claims of the '411 patent.

We affirm the district court's finding that claim 20 of the '411 patent was proven invalid by clear and convincing evidence both as anticipated under 35 U.S.C. § 102 and obvious under 35 U.S.C. § 103. We reverse its findings of invalidity for anticipation for claims 1, 4, and 7-12 because we discern clear error in the district court's finding that the prior art was proven by clear and convincing evidence to have recited every limitation of claims 1, 4, and 7- 12. We also reverse the district court's conclusions that the subject matter of claims 1, 3-13, 15, 17, and 18 of the '411 patent was proven invalid for obviousness by clear and convincing evidence, as we discern clear error in the district court's implicit finding that there was motivation to combine the teachings of the prior art references. As the district court's findings of willful infringement stand unchallenged, we remand for a determination of damages.

BACKGROUND

Edison operates SONGS, a type of nuclear power plant known as a Pressurized Water Reactor ("PWR"). PWRs use water in two systems, called the "primary system" and the "secondary system." A minute amount of water from the secondary system is lost during each operation cycle. This water must be replaced, and the replacement water is commonly referred to as "make-up water." Because water in the secondary system must be of extremely high purity,

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

make-up water is supplied from a make-up demineralization system, which takes outside source water and refines it through "demineralization," *i.e.,* the removal of mineral ions.

During the start-up operations for SONGS, Edison decided to construct a HFMUD to meet the make-up water needs of SONGS' two active reactors. While the HFMUD was being constructed, Edison employed outside vendors, including Ecolochem, to provide the needed make-up water. Before hiring Ecolochem, Edison hired other vendors who provided poor quality water and constantly shuttled demineralization trucks on and off Edison's property to meet Edison's requirements. In the summer of 1982, Edison hired Ecolochem to provide purer quality water at the large volumes Edison needed to meet its make-up water needs. Ecolochem used a patented "Mobile Flow" trailer apparatus to provide water treatment services, which enabled it to regenerate its trailers on-site, thus avoiding the other vendors' needs to truck the impure water off-site. Ecolochem, however, was hired only to produce demineralized water, not water that had been both demineralized and deoxygenated.

The Electrical Power Research Institute ("EPRI"), a research organization for the power industry, published new guidelines in 1982, recommending the use of deoxygenated water in PWRs. These guidelines were soon implemented throughout the utility industry. In direct response to these guidelines, Edison asked Ecolochem to deoxygenate the make-up water used in SONGS. Shortly thereafter, Ecolochem began developing the patented process at issue in the instant case.

***1365** Once construction of the HFMUD was finished, Edison had no further need for Ecolochem's services, being able to now produce, on its own, sufficient, high purity, deoxygenated make-up water by passing the water through the HFMUD. The HFMUD passes the make-up water through a strong acid cation [FN1] resin bed, a predominantly weak base anion [FN2] resin bed, an activated carbon bed, a second strong acid cation resin bed, a strong base anion bed, and, finally, a vacuum deareator. This process removes suspended, undissolved solids and dissolved impurities, including salt, mineral ions, organic chemicals, and oxygen.

> FN1. Cation: An ion having a positive charge. *See Webster's Ninth New Collegiate Dictionary* 216 (1990).
>
> FN2. Anion: An ion having a negative charge. *See Webster's Ninth New Collegiate Dictionary* 87 (1990).

Ecolochem alleged that Edison's process, as described above, infringes its '411 patent. Ecolochem asserted all three independent claims of the '411 patent, and multiple dependent claims. The independent claims read as follows:

> 1. A deoxygenation process comprising a first step of contacting a liquid contacting [sic] dissolved oxygen and hydrazine with a bed of activated carbon to catalyze a reaction between said dissolved oxygen and a portion of said hydrazine, whereby an amount of dissolved carbon contaminants is added to said liquid, and a second step of removing said contaminants and said unreacted hydrazine that comprises passing said liquid through a strong acid cation exchange resin and a strong base anion exchange resin.
>
> 15. A deoxygenation process comprising a first step of contacting water containing dissolved oxygen and hydrazine with a bed of activated carbon to catalyze a reaction between said dissolved oxygen and a portion of said hydrazine, whereby an amount of dissolved and undissolved activated carbon contaminants are added to said water, a second step of removing said dissolved contaminants and said unreacted hydrazine by passing said water through a strong acid cation exchange resin and a strong base anion exchange resin, said water being at a temperature above the freezing point of water and below a temperature that would damage said resins, removing said undissolved contaminants by passing said water through a filter whereby said undissolved contaminants are filtered from said water, and a fourth step of circulating said water in a power generating apparatus after said removing step.
>
> 20. A deoxygenation process comprising a first step of contacting a liquid containing dissolved oxygen and hydrazine with a bed of activated carbon to catalyze a reaction between said dissolved oxygen and said hydrazine, whereby an amount of dissolved activated carbon contaminants is added to said liquid, and a second step of removing at least said dissolved contaminants by passing said liquid through a strong acid cation exchange resin and a strong base anion exchange resin.

Asserted claims 3-13 depend on claim 1. Asserted claims 17 and 18 depend on claim 15. Claim 20 is independent, and no claims dependent thereon are asserted.

All asserted claims in suit recite a deoxygenation process for removing dissolved oxygen from a liquid. In representative claim 20, liquid containing oxygen and hydrazine passes through activated carbon, thus catalyzing a reaction between the oxygen and hydrazine, whereby an amount of dissolved carbon contaminants falls out of the reaction and is added to the liquid. Then the liquid passes through ion exchange resins, including both strong acid cation and strong base anion exchange resins, [FN3] to remove at least the dissolved contaminants. ***1366** The invention of independent claim 1 is similar, except that the resins also remove any unreacted hydrazine. Claims 3

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

through 6 and claims 10 through 13 require further steps in the deoxygenation process, *e.g.*, filtration. Claim 7 requires the temperature of the liquid to be above the liquid's freezing point, and below the temperature at which the resins would be damaged. Claim 8 requires the liquid to be water. Claim 9 requires that the liquid of claim 1 be water and that the water be circulated in a power generating apparatus. Independent claim 15 recites a combination similar to claim 9, and claims 17 and 18 add hydrazine and demineralize the water.

> FN3. A cation/anion exchange resin is a resin which, when brought in contact with liquid containing cation/anion contaminants, removes them.

The '411 patent issued in 1989 from a continuation of an application which issued as the '492 patent on December 5, 1985. Both the '411 and '492 patents are entitled "Deoxygenation Process." At the outset of this litigation, Ecolochem asserted claims under both the '492 and '411 patents.

On September 1, 1994, Judge Gadbois, the original trial judge, granted-in-part and denied-in-part Edison's motion for summary judgment seeking to invalidate various asserted claims of Ecolochem's patents. *See Ecolochem, Inc. v. Southern Cal. Edison Co.*, 863 F.Supp. 1165 (C.D.Cal.1994). Judge Gadbois held claims 1, 2, and 5-10 of the '492 patent and claims 20 and 21 of the '411 patent to be invalid as either obvious and/or anticipated. Ecolochem appealed the partial summary judgment as to claims 1, 2, 5-7, and 10 of the '492 patent and claim 20 of the '411 patent. Ecolochem did not appeal the findings of invalidity of claims 8 and 9 of the '492 patent or claim 21 of the '411 patent. We affirmed the district court's holdings of anticipation of claims 1, 2, 5, and 6 and obviousness of claims 7 and 10 of the '492 patent. We reversed the summary judgment of obviousness of claim 20 of the '411 patent and remanded the case for a trial weighing the secondary consideration evidence as to claim 20. *See Ecolochem, Inc. v. Southern Cal. Edison Co.*, No. 95-1320, 91 F.3d 169, 1996 WL 297601 (Fed. Cir. June 5, 1996) (table).

Upon remand, the case was randomly reassigned to Judge (now Circuit Judge) Richard A. Paez [FN4] to conduct a trial on claim 20 and the remaining claims upon which summary judgment had not been granted in the earlier district court decision. Ecolochem asserted infringement with respect to claims 1, 3-13, 15, 17, 18, and 20 of the '411 patent. *See Ecolochem, Inc. v. Southern Cal. Edison*, No. 92-3436, 1998 WL 1182000, *5 (C.D.Cal.1998) ("*Ecolochem* "). After a bench trial, Judge Paez found that Edison had willfully infringed claims 1, 3-13, 15, 17, 18, and 20 of Ecolochem's '411 patent. Judge Paez, however, held claims 1, 3-13, 15, 17, 18, and 20 of the '411 patent to be invalid for obviousness and claims 1, 4, 7-12 and 20 also to be invalid as anticipated.

> FN4. On March 9, 2000, the Senate confirmed the nomination of Judge Richard A. Paez to the United States Court of Appeals for the Ninth Circuit.

Ecolochem appeals the rulings of the district court that claims 1, 4, 7-12, and 20 were anticipated and that the inventions of claims 1, 3-13, 15, 17, 18, and 20 would have been obvious. Edison does not cross-appeal the district court's findings of willful infringement.

We have jurisdiction under 28 U.S.C. § 1295(a)(1) (1994).

ANALYSIS

I. Anticipation

A. The Martinola Reference

[1][2] The district court found claims 1, 4, 7-12, and 20 of the '411 patent to be anticipated by either of two articles, one published in 1980 and the other in 1981, by Dr. Friedrich Martinola and a co-author, both of which are entitled "Saving Energy by Catalytic Reduction of Oxygen in Feedwater." *See* Dr. Friedrich Martinola & P. Thomas, *Saving Energy by Catalytic Reduction of Oxygen in Feedwater, in Proceedings *1367 of the 41st International Water Conference Pittsburgh* 77 (1980); Dr. Friedrich Martinola and P. Thomas, *Saving Energy by Catalytic Reduction of Oxygen in Feedwater, in Effluent and Water Treatment Journal* 542 (December 1981). [FN5] Because these two articles are in most aspects identical, we refer to them together throughout as "the Martinola reference." We review the district court's decision on anticipation after trial for clear error. *See Lindemann Maschinenfabrik GMBH v. American Hoist and Derrick Co.*, 730 F.2d 1452, 1462, 221 USPQ 481, 488 (Fed.Cir.1984).

> FN5. The International Water Conference is an annual convention focusing on water treatment issues.

The district court focused on Figure 10 of the Martinola reference (which is the same figure in both articles). Figure 10 shows a few possible uses for deoxygenated water, such as supplying make-up water for boilers, or for nuclear reactors. The figure is essentially a flow chart indicating various ways for the water to be deoxygenated, depending on the potential destination of the water. Water designated for nuclear reactors is shown flowing through three steps: demineralization, and then a two-step deoxygenation process, which is accomplished by passing the water through a catalyst column and a mixed bed. [FN6] The district court interpreted Figure 10 as showing that the

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

catalyst column can contain *either* the Lewatit catalyst *or* activated carbon. The district court found that the reactant can be either hydrogen or hydrazine.

> FN6. A mixed bed is a resin bed which, when brought in contact with liquid containing cation and/or anion contaminants, removes them. In other words, a resin bed containing a mixture of cation and anion resins.

On the face of the 1980 Martinola article, which fails to provide a detailed description of Figure 10, there is a bold heading reading: **Application of oxygen reduction in water with hydrogen.** Beneath this heading are two paragraphs of text, followed by Figure 10. The text reads:

> The process is recommended wherever the heat used for thermal degassing cannot be recovered. It will therefore be primarily used where the work has to be carried out at ambient temperature or where there is no heating steam available.
>
> Fig. 10 shows a few suggested applications. It also includes systems for the simultaneous treatment of water with ion exchange resins and oxygen reduction with catalysts.

Martinola at 81-82 (1980). The 1981 article, on the other hand, has no bold headings, but discusses Figure 10 in more detail. The paragraph directly following Figure 10 reads:

> [T]he application of hydrogen to reduce oxygen in water is recommended wherever the heat used for thermal degassing cannot be recovered. It will therefore be primarily used where the work has to be carried out at ambient temperature or where there is no heating steam available.
>
> Figure 10 shows a few suggested applications.

Martinola at 546 (1981). The latter description clarifies that Figure 10 is not meant to illustrate the use of either hydrogen or hydrazine, but only hydrogen, contrary to the findings of the district court in the instant case. This is particularly apparent when we examine the differences between the two articles. The heading in the 1980 article clearly illustrates that Figure 10 refers only to hydrogen, and the discussion in the text explains only that Figure 10 shows a few applications. The 1981 article does not limit Figure 10's applications to hydrogen through a heading, but states clearly in the text ("the application of hydrogen to reduce oxygen.... Figure 10 shows a few suggested applications.") that the suggested applications are for the utilization of hydrogen to reduce oxygen in water.

***1368** We hold that the district court clearly erred in finding that the articles anticipate claims 1, 4, 7-12, and 20 of the '411 patent. Each article is entitled "Saving Energy by Catalytic Reduction of Oxygen in Feedwater." Each article discusses methods to deoxygenate water, dividing the methods in use at the time of the article into physical processes (vacuum degassing at low temperatures and pressure degassing at high temperatures), and chemical processes (reduction with sulfite, hydrazine or hydrogen). The articles state that the physical processes (or thermal degassing) have been the most common method of deoxygenating, as the chemical processes are expensive, slow to react at low temperatures, and contaminate the water by adding salts. The articles then state that the chemical process of deoxygenating water with hydrazine had been in use, but that the chemical process of deoxygenating water with hydrogen had not been, and concludes that the hydrogen process is preferable. [FN7] At the end of the articles is a comparison of the thermal degassing method, the hydrazine method, and the hydrogen method. The articles state that:

> FN7. The '411 specification explains further that "[i]n the prior art deoxygenaton [sic] processes, hydrazine has been used as a strong reducing agent to prevent corrosion and other problems associated with oxygenated water." '411 pat., col. 1, ll. 27-30. A review of the prior art indicates that small amounts of hydrazine were used in the final stage of deoxygenation, after the thermal degassing or other chemical process had been used, to remove the last traces of oxygen, while keeping to a minimum the contaminate byproducts of the hydrazine process.

> Precondition for the proper use of hydrazine is the pH-value of the water to be higher than 8.5, because only in this range the reaction with oxygen takes place with sufficient rate.... When applying activated carbon as a catalyst in the removal of oxygen with hydrazine at ambient temperatures it has to be taken into account that the carbon releases salts into the demineralized water.

Martinola at 81 (1980); Martinola at 545-46 (1981). The articles then conclude that "[i]f we compare the final costs for all three processes ... we find that the method of oxygen reduction with hydrogen is much cheaper than the other methods. The required apparatus is also simple and needs virtually no maintenance." Martinola at 81 (1980); Martinola at 546 (1981). This conclusion is followed by a discussion of how to use water deoxygenated by hydrogen in different industries, as illustrated by Figure 10. Only in that discussion is there a suggestion to follow the catalyst column with a mixed bed if one plans to use water deoxygenated by hydrogen in nuclear power plants. The articles do not discuss the use of water deoxygenated by any other method in nuclear power plants. The articles' discussion of applications of deoxygenated water is limited to the use of hydrogen deoxygenation.

The district court in the instant case disagreed with the earlier discussion of the Martinola reference by the district

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

court in *Ecolochem, Inc. v. Mobile Water Tech. Co.*, 690 F.Supp. 778 (E.D.Ark.1988), *aff'd*, 871 F.2d 1096, 10 USPQ2d 1557, 1989 WL 16031 (Fed.Cir.1989) (table). In the earlier case, the district court held that:

> [t]he diagram found on page 82 of the Martinola article is under the bold- faced heading "Application of oxygen reduction in water with *hydrogen*." [*See* Martinola at 81 (1980)] (emphasis supplied). Thus, the diagram refers to applications of the process the authors are trying to promote--the palladium/hydrogen catalysis--and not the hydrazine process which is discussed in a separate section. This conclusion is buttressed by the fact that the palladium/hydrogen process may also release ionic impurities into the effluent, *see id.* at 79 ("traces of chlorides or other ions may be released"), thus necessitating a downstream ion exchange resin when high purity deoxygenated water is required.... Nothing in the Martinola reference expressly teaches the use of a mixed bed ion exchange ***1369** resin following the hydrazine/carbon process.

Id. at 781-82. This is to be contrasted with the district court in the instant case, which wrote:

> [T]he Court respectfully disagrees with the district court's conclusion in *Mobile Water* that because Diagram 10 is positioned on the page following the heading "Application of oxygen reduction in water with hydrogen" and the associated text, Diagram 10 necessarily refers exclusively to reduction of oxygen in water with hydrogen and not to reduction of oxygen with hydrazine in an activated carbon column.... There is no room for Figure 10 at the bottom of page 81 because Figure 9 extends too far down the page. Following Figure 9 is a brief conclusion to the article, in which the authors recommend use of the Martinola system whenever deoxygenation is to be carried out at ambient temperatures.... Thus, Figure 10 refers generally to methods of deoxygenating and purifying water.

Ecolochem, at *38.

The district court clearly erred by misconstruing Figure 10's relationship to the text of the article. As discussed above, this becomes even clearer once we examine the difference between the two articles. Dr. Martinola and his co-author very carefully made sure in both the 1980 article, through the heading, and the 1981 article, through the language, that Figure 10 refers only to hydrogen and not to hydrazine. The authors would not have been so careful with the language in the 1981 article if they had not meant for Figure 10 to come under the "hydrogen-only" heading in the 1980 article. We disagree with the district court in this case that the publishers rather than the authors chose where to place Figure 10. Rather, we agree with the Arkansas district court that the authors intended the placement of Figure 10 under the hydrogen-only heading, and that Figure 10 therefore only refers to applications of the palladium/hydrogen catalysis. The district court, we hold, clearly erred in finding that Martinola anticipates deoxygenation by hydrazine in combination with a mixed bed. Consequently, we reverse the district court's finding that the Martinola reference was proven by clear and convincing evidence to have anticipated certain of the asserted claims of the '411 patent.

B. The Martinola Presentation

[3] The district court also found the same claims--4, 7-12, and 20--of the '411 patent to be anticipated by a public presentation made by Dr. Martinola at the International Water Conference in Pittsburgh in October 1980. Specifically, the district court found that "Martinola stated at his deposition that during his presentation at the 1980 conference he used Figure 10 of his diagram [sic] as a slide and discussed 'the use of a mixed bed ion exchange resin after hydrazine and activated carbon' for the same purposes and uses described in Ecolochem's patents." *Ecolochem*, at *40 (quoting Martinola Dep. at 18:16-19:6). Finding this to be undisputed, the district court found that this testimony established that the Martinola presentation, like his articles, anticipated the above claims of Ecolochem's '411 patent. *See id.*

[4] Ecolochem argues that "Dr. Martinola's oral presentation must stand or fall with the article as allegedly anticipatory prior art, since Dr. Martinola could not remember in 1992 what he had said twelve years earlier." Appellant's Br. at 35. We do not agree that the presentation, in and of itself, cannot anticipate claims of the '411 patent. Section 102 provides that "a person shall be entitled to a patent unless ... the invention was known or used by others in this country." 35 U.S.C. § 102(a) (1994). A presentation indicative of the state of knowledge and use in this country therefore qualifies as prior art for anticipation purposes under § 102. Furthermore, whether Dr. Martinola correctly remembered his presentation twelve years later is an issue of credibility, on which we review the district court's finding with deference. The district court found, based on ***1370** Dr. Martinola's testimony, that Figure 10 was indeed presented in 1980 by Dr. Martinola with an explanation that it was addressing the use of hydrazine. *See Ecolochem*, at *40.

The key element of the presentation, found by the district court to anticipate the claims of the '411 patent, is Figure 10. We first note that both independent claims 1 and 15 differ from independent claim 20 in contemplating that the final ion exchange step will remove not only dissolved carbon contaminants, but also any excess hydrazine that has not reacted with the dissolved oxygen in the water. Claims 4 and 7-12 (which all depend on claim 1) also require the additional step of removing the excess hydrazine. According to his deposition testimony, Dr. Martinola discussed the removal of the dissolved carbon contaminants at the 1980

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

presentation, but not the removal of excess, *i.e.*, unreacted, hydrazine. [FN8] This omission renders clearly erroneous the district court's finding that claims 1, 4, and 7-12 were anticipated by Dr. Martinola's presentation, and consequently we reverse that finding.

> FN8. Dr. Martinola testified that, during his presentation, he described his paper as showing:
> [T]hat you can apply activated carbon as a catalyst and it will remove oxygen with hydrazine; but it has to be taken into account that the carbon releases salts in the demineralized water, and, afterwards, in Figure 10, there is shown a system with a catalyst column and a mixed bed.
> Martinola Dep. at 18:6-12.

Therefore, the only claim that could be anticipated is claim 20, which claims a deoxygenation process comprising the steps of passing liquid containing oxygen and hydrazine through activated carbon, and then passing the liquid through ion exchange resins, including both strong acid cation and strong base anion exchange resins, to remove at least the dissolved contaminants. This is the exact process described by Dr. Martinola's presentation of Figure 10. We therefore affirm the district court's finding of anticipation of claim 20 of the '411 patent as not clearly erroneous.

II. Obviousness

[5] The district court also held that the inventions of claims 1, 3-13, 15, 17, 18, and 20 of the '411 patent would have been obvious in light of the "combination of the Houghton process for deoxygenation with a mixed bed ion exchange resin to remove excess hydrazine and/or dissolved and/or undissolved carbon contaminants." *Id.* at *37. We review the district court's conclusion of obviousness *de novo*. We affirm the district court's conclusion of obviousness with regard to claim 20, and reverse the holding that obviousness was proven by clear and convincing evidence with regard to all other claims at issue.

A. The Houghton Process as a "Blueprint"

[6] The district court essentially found that the most innovative aspect of Ecolochem's process was its "[u]nearthing [of] long-neglected art," holding that "Ecolochem's good fortune in obtaining the Houghton reference just as the EPRI guidelines created increased attention in the PWR industry to the problem of ambient temperature deoxygenation does not entitle Ecolochem to patent protection." *Id.* at *33.

Houghton was the co-author of a paper on deoxygenation by carbon catalysis of the reaction between hydrazine and dissolved oxygen, entitled "The Use of Active Carbon With Hydrazine in the Treatment of Boiler Feed Water" ("the Houghton article"). The Houghton article was presented at the 1957 International Water Conference at Bournemouth, England and brought to the attention of Ecolochem at the 1982 International Water Conference. [FN9] ***1371** The Houghton article peaked the interest of Ecolochem, and its scientists began performing experiments to determine if the results predicted in Houghton's paper could be replicated in a laboratory setting. Ecolochem's scientists were skeptical, because they "had understood that hydrazine reacted very slowly with dissolved oxygen and one of [their] objectives in [the] preliminary experimentation was to run the process to determine if the Houghton process sufficiently catalyzed the reaction." Miller Decl. ¶ 22. The tests supported the Houghton article, but also revealed the presence of ionic substances in the deoxygenated water coming out of the carbon bed. Houghton did not discuss the ionic contamination. Ecolochem performed subsequent tests to assess its attempts to reduce the presence of the ionic substances, and after considerable experimentation, eventually succeeded with the patented process after more research and considerable experimentation. The district court found that "for years it had been known in the art of water treatment that activated carbon releases ionic substances into water," but that Ecolochem's scientists, who the district court found had been employed in the water treatment industry for over a decade by the time the patent issued, were unaware of this contamination at the time they conducted their experiments. *Ecolochem.* at *11.

> FN9. William Miller, one of the inventors of the deoxygenation process described in the '411 patent, testified that he attended the International Water Conference in 1982 "with the intention of finding alternative deoxygenation processes, if there were any." Miller Decl. ¶ 19 (Oct. 28, 1997), J.A. at 931. By alternative, Mr. Miller was referring to processes other than deoxygenation through the use of a vacuum degasifier. While at the conference, Mr. Miller spoke to an Ecolochem area sales manager, John Pugsley.
> Mr. Pugsley worked for the Florida Power and Light Company before he came to work for Ecolochem and he recalled that someone at Florida Power and Light was doing some work on deoxygenation involving carbon. As a result of that conversation, Mr. Pugsley arranged for someone at Florida Power and Light to send me [the Houghton article].
> *Id.*

The district court recognized that the Houghton reference did not anticipate the patent, but felt that:

> Taken together, the prior art references relevant to Ecolochem's invention disclose all of the elements of the claimed invention, and their combined teachings would

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

> have suggested to one of ordinary skill in the art that the Houghton process could be followed by the use of mixed bed ion exchange resins to provide ambient temperature deoxygenation and remove excess hydrazine as well as dissolved and undissolved carbon contaminants.

Id. at *34.

"A patent may not be obtained ... if the differences between the [claimed invention] and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art." 35 U.S.C. § 103(a) (Supp. III 1997). Our analysis of the patentability of Ecolochem's invention begins with the phrase "at the time the invention was made." Here, the date of the invention is presumed to be the filing date of the parent application, December 16, 1983.

[7] In *In re Dembiczak,* we noted that:

> Measuring a claimed invention against the standard established by section 103 requires the oft-difficult but critical step of casting the mind back to the time of invention, to consider the thinking of one of ordinary skill in the art, guided only by the prior art references and the then-accepted wisdom in the field.

In re Dembiczak, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed.Cir.1999). We "cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention." *In re Fine,* 837 F.2d 1071, 1075, 5 USPQ2d 1596 (Fed.Cir.1988).

Our case law makes clear that the best defense against hindsight-based obviousness analysis is the rigorous application of the requirement for a showing of a teaching or motivation to combine the prior art references. *See Dembiczak,* 175 F.3d at 999, 50 USPQ2d at 1617. "Combining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art ***1372** to defeat patentability--the essence of hindsight." *Id.*

[8][9] "When a rejection depends on a combination of prior art references, there must be some teaching, suggestion, or motivation to combine the references." *In re Rouffet,* 149 F.3d 1350, 1355, 47 USPQ2d 1453, 1456 (Fed.Cir.1998) (citing *In re Geiger,* 815 F.2d 686, 688, 2 USPQ2d 1276, 1278 (Fed.Cir.1987)). The same principle applies to invalidation. "Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination." *ACS Hosp. Sys., Inc. v. Montefiore Hosp.,* 732 F.2d 1572, 1577, 221 USPQ 929, 933 (Fed.Cir.1984). Although the suggestion to combine references may flow from the nature of the problem, *see Pro-Mold & Tool Co. v. Great Lakes Plastics, Inc.,* 75 F.3d 1568, 1573, 37 USPQ2d 1626, 1630 (Fed.Cir.1996), "[d]efining the problem in terms of its solution reveals improper hindsight in the selection of the prior art relevant to obviousness," *Monarch Knitting Mach. Corp. v. Sulzer Morat Gmbh,* 139 F.3d 877, 880, 45 USPQ2d 1977, 1981 (Fed.Cir.1998). Therefore, "[w]hen determining the patentability of a claimed invention which combines two known elements, 'the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination.' " *In re Beattie,* 974 F.2d 1309, 1311-12, 24 USPQ2d 1040, 1042 (Fed.Cir.1992) (quoting *Lindemann,* 730 F.2d at 1462, 221 USPQ at 488).

[10] In this case, the district court used the '411 patent as a blueprint, with the Houghton process as the main structural diagram, and looked to other prior art for the elements present in the patent but missing from the Houghton process. The district court opinion does not discuss any specific evidence of motivation to combine, but only makes conclusory statements. "Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence.' " *Dembiczak,* 175 F.3d at 999, 50 USPQ2d at 1617. The district court provides no support for its broad conclusory statement that it was known in the art that a carbon bed, as used in the Houghton process, would produce water with high levels of conductivity caused by the presence of ionic contaminants. Nor does the district court then provide support for its implicit finding that given water so contaminated, it would be obvious to one of ordinary skill in the art to place a mixed bed ion exchange resin downstream of the carbon bed. In fact, nowhere does the district court particularly identify any suggestion, teaching, or motivation to combine the Houghton process with a mixed bed ion exchange resin to achieve the patented process.

The district court avoids the issue, and makes implicit findings, but can point to nothing that suggests the combination of deoxygenation and demineralization processes that comprise Ecolochem's invention. For instance, the district court finds that Ecolochem "did not try any other way to remove the ionic contaminants leached by the carbon bed other than adding the mixed bed," *Ecolochem,* at *12, and that "[t]here is no evidence that any system for production of ultra-pure water ever included a carbon bed as the final step in water treatment." *Id.* at *22. In addition, the district court makes the unsupported finding that "[m]any in the art knew as of 1982 that carbon beds leached contaminants which could be removed by ion exchange." *Id.* at * 21. In support of this statement, the district court relies upon the Martinola reference, and U.S. Patent No. 4,430,226 ("the '226 patent") for a disclosure that "activated carbon leaches dissolved contaminants." For the

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

disclosure of the "removal of dissolved carbon contaminants by ion exchange", the district court relied upon the '226 patent and John W. Hassler's 1974 article "Purification with Activated Carbon," which do state that with some types of carbon, "ion-exchange resins have been employed to remove inorganic compounds, alkalinity, or acidity not absorbable by activated carbon." J.A. at ***1373** 1893. However, while these references teach the leaching of dissolved contaminants by activated carbon and the use of ion-exchange resins to remove carbon contaminants, neither reference suggests combining, nor provides any motivation to so combine, the two elements of the Ecolochem process, *i.e.,* deoxygenation of the water by the Houghton process and demineralization of the water by the mixed bed.

The district court seems to find that the Martinola reference implicitly suggests the combination of the two elements, but discounts "[t]he fact that Martinola did not make Ecolochem's invention, and instead focused on the hydrogen-palladium method of deoxygenation [a]s not relevant." *Id.* at *39. The district court clearly erred in this regard. This fact is completely relevant to the obviousness analysis, since Martinola actually teaches away from combining at least one of the Martinola articles with the Houghton process to achieve Ecolochem's claimed process. While the Martinola reference describes a hydrogen and Lewatit-based deoxygenation process and mentions deoxygenation by carbon catalysis of a hydrazine/oxygen reaction, it does so only for comparative purposes. The Martinola reference actually unfavorably compares the hydrazine/carbon process, saying that it "releases salts into the demineralized water" and that the hydrogen-based process is energy saving and significantly less expensive. [FN10]

> FN10. The 1980 article states:
> A large number of chemical compounds can be used for reducing the elementary oxygen in water.... Until now, however, they have merely been used for the residual degassing that follows on from thermal processes. The reasons for this are the high costs involved, the slowness to react at low temperatures and the additional introduction of salts into the water. For the elimination of oxygen in boiler feedwater and hot water, use has been made for a long time now of hydrazine in the form of hydrazine hydrate.... A particularly simple process, which takes place in particularly well-known stages and is also energy-saving, is the reaction between the dissolved oxygen and hydrogen gas introduced into the water.... If we compare the final costs for [the different methods] we find that the method of oxygen reduction with hydrogen is much cheaper than the other methods. The required apparatus is also simple and needs virtually no maintenance.
> Martinola at 77-81 (1980).

The Martinola reference is not the only reference that points to problems and concerns with the Houghton process. The Houghton article was challenged from the day it was presented, when an audience member asked whether the process could produce water with acceptable amounts of silica. Houghton responded that contamination was a concern with his process, and both this inquiry and the response were published with the paper in a section entitled "Discussion." Another paper published contemporaneously with the Houghton article states that:

> if an activated carbon-bed is used to accelerate the oxygen/hydrazine reaction, it is particularly important that no trace of carbon should enter the boiler ... [i]f it is considered essential that the hydrazine and oxygen should react before entering the boiler, then the use of ultra-violet light is a promising means of accelerating the reaction, without introducing any impurities into the system.

S.R.M. Ellis, C. Moreland, *The Reaction Between Hydrazine & Oxygen, in The Account of the Proceedings of the International Conference held at Bournemouth, 15th-17th May 1957* 8, 21 (1958). In 1960, T.F. Demmitt prepared a "Preliminary Report on the Use of Activated Carbon as a Catalyst for the Dissolved Oxygen-Aqueous Hydrazine Reaction." Demmitt stated that "[m]agnetite would be more desirable than activated carbon since there would be no tendency to 'deactivate' magnetite in filtered water," and thereby taught away from the idea of using a carbon bed as the catalyst. J.A. at 1696H. In 1962, a paper was presented at the International Water Conference, stating that research had shown two methods "to remove oxygen to sufficiently low levels with adequate capacities ***1374** to be practical." Piero Sturla, *Polishing Condensate and Dearating by Ion Exchange* at 63 (1962). Neither of these methods was based on the Houghton process. These two methods were still being used in 1977, when Culligan (a water treatment company) recommended one of the two in an internal memo for use in portable containers, and the challenges continued through the time of the invention. Even Ecolochem's scientists themselves testified that, prior to their successful tests, they did not believe they would be able to replicate the results stated in the Houghton article.

Furthermore, the district court found even the mixed bed, of which the use "to remove carbon contaminants was well known in the field," had detractors. *Ecolochem, at* *39. Edison's own engineers testified that they considered:

> a four-bed [a primary cation bed, a primary anion bed, a secondary cation bed and a secondary anion bed] system superior to ... a three-bed system [a cation bed, an anion bed and a mixed bed] ... because of the problems that Edison had encountered in regenerating the mixed bed, ... and based on ... personal experience that four-bed systems

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

generally outperformed mixed bed systems in producing pure water.

Id. at *13. There is clear evidence of teaching away in the prior art from both the demineralization process and the deoxygenation process used by Ecolochem, and no evidence that there was any suggestion in the prior art to combine these two processes, yet the district court finds the '411 patent obvious in light of the prior art.

The absence of a convincing discussion of the specific sources of the motivation to combine the prior art references, particularly in light of the strength of prior art teaching away from the use of the Houghton process, is a critical omission in the district court's obviousness analysis, which mainly discusses the ways that the multiple prior art references can be combined to read on the claimed invention. For example, the district court finds that the invention of claim 20 would have been obvious, and that, although claims 1 and 15 differ from claim 20, "[e]ach of the additional steps of claims 1 and 15 is disclosed in the prior art." *Id.* at *22. The opinion then lists each step and states where in the cited prior art references the step can be found. This reference-by-reference, limitation-by-limitation analysis wholly fails to demonstrate how the prior art teaches or suggests the combination claimed in the '411 patent.

With hindsight, we could perhaps agree that the Houghton article seems like an obvious place to start to address the need in the power plant industry for an improved carbon-catalyzed deoxygenation process employing hydrazine that can be used commercially in a variety of applications. But, "obvious to try" is not the standard. As embodied in the '411 patent, the process would secure for the art all of the advantages that catalyzation of a reaction normally provides without prohibiting its use due to all of the incumbent disadvantages associated with the use of hydrazine. However, the incumbent disadvantages associated with the use of hydrazine to deoxygenate water as described in the Houghton article include the release of unwanted impurities into the water, including dissolved substances such as salts, organic chemicals and suspended solids. As a result, the industry widely regarded the Houghton process as too impractical to be used for large amounts of water, or for high purity deoxygenated water. Under the conditions in which PWRs operate, even minute amounts of these contaminants can adversely affect generator integrity. *See id.* at *5. Variations on this method had been tried, but discounted in favor of other deoxygenation processes. Dr. Martinola himself testified that at the time of his presentation he stated that the hydrazine method of deoxygenation led to contaminated deoxygenated water. *See* supra at I.B. Finally, the process not only worked, but worked better than expected, leading Edison to investigate "why Ecolochem's equipment can reduce [the organic concentration in the water] to less than 10 ***1375** p[arts] p[er] b[illion] with a device not designed to [do so]." *Ecolochem,* at *15. Edison, in fact, gave four prospective bidders for the HFMUD construction job a tour of Ecolochem's equipment and shared "information about Ecolochem's process with others in violation of [Ecolochem's] secrecy agreement." *Id.* Only after this sharing of information and subsequent testing was a prospective bidder able to develop the process used in the HFMUD, and accused by Ecolochem of infringing the ' 411 patent. The district court in fact found that the bidder's "proposed system ... effectively copied Ecolochem's deoxygenation process." *Id.* at * 15. This evidence undermines the district court's conclusion that Ecolochem's process would have been obvious to one of ordinary skill in the art.

Because we do not discern any evidentiary basis for the finding by the district court that there was a suggestion, teaching, or motivation to combine the prior art references cited against the claimed invention, the district court's conclusion of obviousness cannot stand. The implicit generalized finding by the district court that, when one of ordinary skill in the art was faced with the problem of deoxygenating water for use in a nuclear power plant and the Houghton article, the combination claimed by Ecolochem in the '411 patent would have been obvious is insufficient. We have previously held that "[t]he suggestion to combine may be found in explicit or implicit teachings within the references themselves, from the ordinary knowledge of those skilled in the art, or from the nature of the problem to be solved." *WMS Gaming, Inc. v. International Game Tech.,* 184 F.3d 1339, 1355, 51 USPQ2d 1385, 1397 (Fed.Cir.1999). However, there still must be evidence that "a skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed." *In re Rouffet,* 149 F.3d at 1357, 47 USPQ2d at 1456; *see also In re Werner Kotzab,* 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed.Cir.2000) ("[A] rejection cannot be predicated on the mere identification ... of individual components of claimed limitations. Rather, particular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed."). Here, there was no such evidence presented. The only evidence on this issue presumes the very problem at hand--two experts testified that "*if* someone of ordinary skill in the art had been given the Houghton reference in 1982 and [*if*] they were asked to make it usable in a high-pressure power plant, they would have come up with Ecolochem's invention." *Ecolochem,* at *31 (emphasis added). The evidence available, however, indicates that if one of ordinary skill in the art had been given the Houghton

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

reference, they would not have been inclined to use it, due to the large amount of teaching away, and the reliance in the industry on vacuum degasifiers to deoxygenate water. This finding by the district court presumes the knowledge acquired from Ecolochem's patent. We hold that the district court's finding that a skilled artisan would combine these references was clearly erroneous, and we hold that on this record the district court clearly erred in finding clear and convincing evidence of a suggestion to combine the prior art references, a suggestion to use the Houghton article as the backbone of the invention. Therefore, we reverse the district court's conclusion of obviousness with regard to claims 1, 3-13, 15, 17, and 18 of the '411 patent. As to claim 20, however, our prior decision mandates that we now undertake further analysis respecting its invalidation by the district court.

B. *Prima Facie* Case of Obviousness of Claim 20

[11] We previously held in our June 1996 non-precedential decision that:

> the district court clearly mischaracterized the import of Demmitt as a prior ***1376** art reference for the determination of obviousness of claim 20. Demmitt ... did not disclose the removal of carbon contaminants with an ion exchange resin. Despite the district court's mischaracterization of the importance of Demmitt, Ecolochem concedes that there is a 'prima facie case of obviousness before Demmitt and there remains one after.' However, Ecolochem argues that the secondary considerations, in this case, could rebut the prima facie case and that the district court wrongly failed to consider its evidence of secondary considerations and conclude that the evidence raised a genuine issue of fact requiring trial. We agree.... For the foregoing reasons ... the trial court's holding of obviousness of claim 20 of the '411 patent is reversed, and the case is remanded for trial on validity and infringement as to claim 20.

Ecolochem, at *4-5. Given our earlier reliance on Ecolochem's concession of the *prima facie* obviousness of claim 20, which is binding as law of the case, we must treat claim 20 separately from the other claims held obvious in the decision appealed herein. We may only conclude that Ecolochem rebutted the *prima facie* case of obviousness with regard to claim 20 if the evidence of secondary considerations supports such a holding. *See id.; In re Piasecki,* 745 F.2d 1468, 1471, 223 USPQ 785, 787 (Fed.Cir.1984). We therefore review the district court's findings on the evidence of secondary considerations, and, for purposes of completeness, address whether and how this evidence affects the adjudicated invalidity of all other asserted claims.

C. Evidence of Secondary Considerations With Regard to All Claims

[12] The idea that a patented invention might appear to be obvious given the excellent vision accorded by hindsight, but might not have been obvious at the time the invention was made, was discussed by the Supreme Court in *Graham v. John Deere Co.,* 383 U.S. 1, 17-18, 86 S.Ct. 684, 15 L.Ed.2d 545 (1966). In the intervening years since *Graham* a great deal of attention has been paid to the importance of secondary considerations. We discuss below many of the secondary considerations used by the courts in an effort to compensate for hindsight. The Supreme Court held that "[s]uch secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy." 383 U.S. at 17-18, 86 S.Ct. 684 (internal citations omitted). While we review the district court's factual findings on the secondary considerations for clear error, we review the ultimate determination of obviousness *de novo. See B.F. Goodrich Co. v. Aircraft Braking Sys. Corp.,* 72 F.3d 1577, 1582, 37 USPQ2d 1314, 1317-18 (Fed.Cir.1996). We find clear error in the district court's findings on several of the secondary considerations as discussed below, but upon reviewing the obviousness conclusion *de novo,* hold that even if corrected, the findings respecting secondary considerations support the district court's holding of obviousness of claim 20 in the instant case. This conclusion has no effect on the obviousness of the inventions of claims 1, 3- 13, 15,17, and 18, which we have already held would not have been obvious given the absence of any motivation to combine the teachings of the cited prior art references. Ecolochem has, however, failed to rebut the conceded *prima facie* case of obviousness with regard to claim 20. We therefore affirm the district court's conclusion that the invention of claim 20 would have been obvious.

1. Long-Felt But Unsolved Need

The district court found that "[t]he findings of fact do not support Ecolochem's contention that there was a long-felt but unsolved need to create an ambient temperature deoxygenation process for use in start up and restart of nuclear power facilities." ***1377** *Ecolochem,* at *35. It went on to hold that even if there had been such a need, the "required nexus between the claimed invention and the long-felt need is attenuated by the increased market demand resulting from adoption of the EPRI guidelines in 1982." *Id.*

Ecolochem argues that the 1982 EPRI guidelines merely reflected the acknowledged need for deoxygenated water that arose in the late 1970s and that the regulations only recommended, rather than required, deoxygenated water. Ecolochem further argues that there were reasons other than the issuance of these regulations for it to develop its

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

processes. Furthermore, it argues that its mobile chemical process was the only way to meet this long-felt but unsolved need for deoxygenated water during the time required to build and install the permanent deoxygenators. Ecolochem argues that the guidelines (like emissions requirements for automobiles) were issued only after a long- felt need was realized, and that the regulations may have been unattainable at the time they were issued. Even Edison in its brief pointed out that Ecolochem had the only practical solution to the need of SONGS for deoxygenated and demineralized water before the completion of the HFMUD.

Our review of the district court's finding of no long-felt but unsolved need is for clear error, a deferential standard of appellate review. We cannot say the district court's finding that Ecolochem produced demineralized water in response to the guidelines issued by EPRI to be clearly erroneous. The record shows that the guidelines urging the use of deoxygenated water were issued in 1982, and Ecolochem filed for a patent on its deoxygenization process on December 16, 1983. This evidence supports the district court's finding that Ecolochem's process was developed not in response to a long-felt need in the power industry, but in response to a shortly-felt requirement imposed by EPRI's guidelines.

2. Commercial Success

[13] We have previously held that a party cannot "demonstrate commercial success, for purposes of countering the challenge of obviousness, unless it can show that the commercial success of the product results from the claimed invention." *J.T. Eaton & Co. v. Atlantic Paste & Glue Co.*, 106 F.3d 1563, 1571, 41 USPQ2d 1641, 1647 (Fed.Cir.1997). We have further held that a presumption arises that the patented invention is commercially successful "[w]hen a patentee can demonstrate commercial success, usually shown by significant sales in a relevant market, and that the successful product is the invention disclosed and claimed in the patent." *Id.* In the instant case, Ecolochem offered evidence that its invention was practiced at 28 plants and generated almost $13 million in revenue from 1983 to 1990. Once Ecolochem made the requisite showing of nexus between commercial success and the patented invention, the burden shifted to Edison to prove that the commercial success was instead due to other factors extraneous to the patented invention. *See id.*

At trial, Edison countered that the commercial success of Ecolochem's process was due solely to the fact that it was part of a mobile apparatus, and that since none of the claims at issue include such a mobility limitation, the commercial success factor favors a finding of obviousness in the instant case. The district court found that both Edison and Ecolochem had carried their respective burdens, finding that the volume of Ecolochem's sales satisfied the requirements to show commercial success, but also that Edison proved the commercial success was due to factors other than those claimed. The district court explained:

> Ecolochem's limited commercial success with its patented process was primarily attributable to (1) Ecolochem's ability to meet the need in the PWR industry for short-term emergency services through its Mobile Flow service; (2) Ecolochem's ***1378** ability to provide deoxygenation services while PWR plants built permanent deaeration equipment to meet new EPRI standards; (3) Ecolochem's head start in marketing mobile deoxygenation treatment services through use of its patented Mobile Flow trailer; (4) the increased blowdown rates that resulted from the 1982 tightening of EPRI guidelines for salts in the steam generator. In short, Ecolochem's commercial success was due not to the nature of the claimed invention, but to other economic and commercial factors unrelated to the technical quality of the patented process.

Ecolochem, at *36.

The district court's finding, however, ignores Edison's own statements to the contrary. Edison indicated in its briefs that it chose Ecolochem because the "first vendors Edison hired provided poor quality water and constantly shuttled demineralization trucks on and off Edison's property to meet Edison's capacity requirements." Appellee's Br. at 3. Edison's brief also states that Ecolochem's process "enabled it, unlike Edison's previous vendors, to regenerate its trailers on-site and provide purer quality water at the large volumes Edison needed." *Id.*

These statements make it clear that the commercial success of Ecolochem's product was, in fact, based on two factors: the improved filtration process, and the mobility of the commercial embodiment. Edison did not differentiate the improved filtration process from the patented process in any way, focusing only on the missing limitation of mobility in the claims at issue. The success was due to both the mobility, undisputedly not covered by the claims, and to the improved filtration process, undisputedly covered by the claims. Edison had the burden of disproving that the improved filtration process contributed to the success of the invention, and its own brief undermines its argument, by conceding the benefits of Ecolochem's filtration process. *See Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 716, 21 USPQ2d 1053, 1055 (Fed.Cir.1991) ("To overturn a patent, the challenger must clearly prove those facts which support patent invalidity."). Consequently, the district court clearly erred in finding that Edison met its burden of proving that the success of the invention was due to factors not claimed in the patented invention.

3. Failure of Others to Make Invention

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

In evaluating this factor, the district court held that an unsolved problem is not evidence of non-obviousness unless skilled workers in the art have tried and failed to solve the problem. *See Ecolochem,* at *36. The district court then held that Ecolochem did not show that others had tried but failed to make Ecolochem's invention. *See id.*

Ecolochem cited a 1986 internal business memorandum circulated at Arrowhead, a competitor of Ecolochem's. In this memorandum, under the heading "*OPPORTUNITY,*" Arrowhead wrote the following: "Ecolochem claims a proprietary process for deoxygenation and has exploited this market with little or no interference from Arrowhead. Markets exist primarily at the nuclear plants. Several approaches for deoxygenation exist...." J.A. at 3049.

[14] Edison, in turn, responded with testimony from its employees that when Edison discussed its deoxygenation needs with contractors in 1982, only two contractors mentioned chemical deoxygenation: Arrowhead and Ecolochem. Edison employees further testified that vacuum degasifiers were the industry standard, were reliable and economical, and that every other contractor stated that it intended to deoxygenate with a vacuum degasifier. There was no testimony as to whether Arrowhead succeeded in making a working chemical deoxygenation device. The district court evaluated Edison's testimony, and the absence of any testimony contradicting it, and found that no competitor had attempted to employ a chemical deoxygenation process. The failure of others is therefore reduced to a credibility question, whether the district court believed the Edison employees. "This court gives great ***1379** deference to the district court's decisions regarding credibility of witnesses." *Carroll Touch, Inc. v. Electro Mechanical Sys., Inc.,* 15 F.3d 1573, 1580, 27 USPQ2d 1836, 1842 (Fed.Cir.1993) (*citing Anderson v. City of Bessemer City,* 470 U.S. 564, 575-76, 105 S.Ct. 1504, 84 L.Ed.2d 518 (1985)). We cannot find it clearly erroneous on the part of the district court to have found no evidence of failure by others to make a chemical deoxygenation device or develop such a process.

4. Simultaneous Invention

[15] "The fact of near-simultaneous invention, though not determinative of statutory obviousness, is strong evidence of what constitutes the level of ordinary skill in the art." *The Int'l Glass Co. v. United States,* 187 Ct.Cl. 376, 408 F.2d 395, 405 (1969). "[T]he possibility of near simultaneous invention by two or more equally talented inventors working independently, ... may or may not be an indication of obviousness when considered in light of all the circumstances." *Lindemann,* 730 F.2d at 1460, 221 USPQ at 487.

The district court found that "prior to the issuance of Ecolochem's patent, Todd Hook of NWT independently combined the Houghton process with a mixed bed ion exchange resin, thereby independently developing Ecolochem's invention." *Ecolochem,* at *36. Ecolochem argues that Mr. Hook testified at trial that the:

> idea embodied in his test apparatus, which combined the hydrazine/carbon process and ion exchange, was not his but Dr. Sawochka's. Dr. Sawochka, an acknowledged expert in the field, provided the system sketch and configured the test apparatus embodying the idea. Obviousness or nonobviousness to experts is irrelevant to obviousness under Section 103. Competing innovation by Dr. Sawochka may show bias but nothing more.

Appellant's Br. at 56. Edison responds by stating that "as both Sawochka and Hook testified, Hook was left largely on his own in performing his project." Appellee's Br. at 53.

The issue of simultaneous invention is directly tied to the level of knowledge attributable to one of ordinary skill in the art. The district court weighed the testimony discussed above, and determined that there was evidence of simultaneous invention, and the fact that the one who performed the experiment was supervised by one of extraordinary skill in the art did not enhance the former's level of knowledge. Essentially, the district court found that this secondary consideration factor favors obviousness. As this was based on the district court's determination that Dr. Sawochka's and Mr. Hook's testimony was credible, we must give that finding great deference. Consequently, we cannot discern clear error in the finding of simultaneous invention by the district court.

5. Teaching Away

[16] The district court found that "Ecolochem presented no evidence that the prior art expressed skepticism concerning the efficacy of using the Houghton process or of combining the Houghton process with ion exchange. Likewise, the Court's findings of fact establish that the prior art did not teach away from Ecolochem's invention." *Ecolochem,* at *37. Ecolochem argued that the Houghton process had been around for decades and was not well known or regarded, citing a number of articles stating concerns with the Houghton process.

We previously found, in our unpublished June 1996 decision, that Ecolochem had put forth evidence "that various references taught away from the invention as they warned against the Houghton process because of the carbon contaminants." *Ecolochem, Inc. v. Southern Cal. Edison Co.,* 1996 WL 297601, at *5. We discussed above the fact that Martinola teaches away from the Houghton process, as that reference found the process inefficient and expensive. We also discussed above the history of prior art teaching away from Houghton. Based on the body of evidence

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

provided to the district court, we find ***1380** clearly erroneous the district court's determination that the prior art did not teach away from using the Houghton process or from combining it with a mixed bed in any manner.

6. Copying and Acclamation

[17] Another indicia of non-obviousness of a product is the acclamations it receives when it is released, and the copying that occurs. The district court found "that [although] Ecolochem's process was warmly received in the water treatment industry and was copied by at least two competitors, that copying did not result from the novelty of Ecolochem's invention and is of little weight in assessing obviousness." *Ecolochem,* at *37.

[18] Ecolochem argues persuasively that its device was copied because it was significantly less expensive than vacuum degasifiers, the only alternative in the field. Furthermore, testimony from Edison indicates that Ecolochem's process was better than its competitors'. The district court gives little credit to Ecolochem's arguments, finding only that the process was not novel. It is a factual determination as to what the exact reason for the copying was, but here we hold that it was clear error on the district court's part to discount the copying because the court believed that the process was not novel. Therefore, this factor cuts against a conclusion of obviousness.

[19] We note, however, that a showing of copying is only equivocal evidence of non-obviousness in the absence of more compelling objective indicia of other secondary considerations. *See In re GPAC,* 57 F.3d 1573, 1580, 35 USPQ2d 1116, 1122 (Fed.Cir.1995) (" '[M]ore than the mere fact of copying by an accused infringer is needed to make that action significant to a determination of the obviousness issue.' ") (quoting *Cable Elec. Prods. v. Genmark, Inc.,* 770 F.2d 1015, 1028, 226 USPQ 881, 889 (Fed.Cir.1985)). The reason is that the alleged copying "could have occurred out of a general lack of concern for patent property." *See Cable Elec. Prods.,* 770 F.2d at 1028, 226 USPQ at 889. Consequently, even though the district court clearly erred in stating that the copying was not connected to the patented aspects of the invention, its error does not carry great weight in our evaluation of the obviousness of the invention of claim 20 in light of all the secondary considerations, combined with the other evidence and findings on the prior art.

7. Weighing of the Factors

We must now weigh all the secondary considerations to determine whether the district court's error in not finding commercial success and teaching away, when aggravated by its error in failing to find copying; renders erroneous its conclusion that the invention of the claims at issue here would have been obvious. The district court did not clearly err in finding that there was no long-felt but unsolved need, that there was no failure by others to make the invention, and that there was a successful simultaneous invention by others. As explained above, the district court clearly erred in finding that the commercial success of Ecolochem's product was based only upon its mobility; that the prior art did not teach away from the Houghton process; and that there was no evidence of copying. However, only two of these factors are actually stated in *Graham;* the other falls under *Graham* 's "etc." clause. The factors specifically mentioned in *Graham,* and those that we give the most weight to in the instant case, are the commercial success of the invention, long-felt but unsolved needs, and failure of others to invent.

In our *de novo* obviousness review, we hold that the absence of any evidence that others were trying to emulate the patented process, and the fact that within two years of the publication of the EPRI guidelines Ecolochem had applied for a patent on the process, is more indicative of obviousness than the evidence of modest commercial success, and the teaching away from the use of the unmodified Houghton process is indicative of non-obviousness. The question of obviousness in the instant case is a ***1381** close one, but we hold that the secondary considerations, taken as a whole, do not overcome the other evidence of obviousness.

Our earlier unpublished decision placed the burden on Ecolochem to rebut, based on all the secondary considerations, the *prima facie* case of obviousness with regard to claim 20. We cannot say that the district court erred in concluding that Ecolochem has not rebutted the *prima facie* case of obviousness with regard to claim 20.

However, for the remaining claims at issue, claims 1, 3-13, 15, 17, and 18, the secondary consideration evidence is but one of several factors, all of which must be assessed in determining obviousness. *See Graham,* 383 U.S. at 17-18, 86 S.Ct. 684. Given the absence of any proof of a motivation to combine, we hold that the remaining claims were not proven obvious by clear and convincing evidence in light of the prior art.

CONCLUSION

For the reasons stated above, we affirm the district court's holding that it was proven by clear and convincing evidence that the invention of claim 20 would have been obvious and was anticipated, but reverse the district court's holdings that claims 1, 3-13, 15, 17, and 18 were proven invalid either as anticipated and/or obvious. We remand this case for award of damages, consistent with the district court's finding of willful infringement which stands. The decision, therefore, is

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

AFFIRMED-IN-PART, REVERSED-IN-PART, and REMANDED.

COSTS

Defendant shall pay costs.

227 F.3d 1361, 56 U.S.P.Q.2d 1065

END OF DOCUMENT

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

United States Court of Appeals,
Federal Circuit.

CROWN OPERATIONS INTERNATIONAL, LTD. and Marshall H. Krone, Plaintiffs-Appellants,
v.
SOLUTIA INC., Defendant-Appellee.

No. 01-1144.

DECIDED: May 13, 2002.
Rehearing Denied: June 10, 2002.

Competitor of holder of patents for layered films used to create safety and solar control glass brought suit seeking declaratory judgment that patents were invalid. The United States District Court for the Western District of Wisconsin, John C. Shabaz, J., granted summary judgment denying relief. Competitor appealed. The Court of Appeals, Gajarsa, Circuit Judge, held that: (1) two percent limitation for visible reflectance contribution that was claimed in first patent was not inherent in, and thus was not anticipated by, existing patent; and (2) first patent was not invalid for obviousness; but (3) fact issue as to whether second patent satisfied enablement requirement precluded summary judgment.

Affirmed in part, reversed in part, and remanded.

West Headnotes

[1] Federal Courts ⚖776
170Bk776 Most Cited Cases

Court of Appeals reviews a district court's grant of summary judgment without deference. Fed.Rules Civ.Proc.Rule 56(c), 28 U.S.C.A.

[2] Federal Civil Procedure ⚖2543
170Ak2543 Most Cited Cases

On a motion for summary judgment, the evidence must be viewed in the light most favorable to the party opposing the motion, with doubts resolved in favor of the nonmovant. Fed.Rules Civ.Proc.Rule 56(c), 28 U.S.C.A.

[3] Federal Civil Procedure ⚖2546
170Ak2546 Most Cited Cases

Once party moving for summary judgment has satisfied its initial burden, the opposing party must establish a genuine issue of material fact and cannot rest on mere allegations, but must present actual evidence. Fed.Rules Civ.Proc.Rule 56(c), 28 U.S.C.A.

[4] Federal Civil Procedure ⚖2470.1
170Ak2470.1 Most Cited Cases

Issues of fact are genuine, and thus sufficient to preclude grant of summary judgment, only if the evidence is such that a reasonable jury could return a verdict for the nonmoving party. Fed.Rules Civ.Proc.Rule 56(c), 28 U.S.C.A.

[5] Federal Civil Procedure ⚖2470.1
170Ak2470.1 Most Cited Cases

A disputed fact is material, so that summary judgment may not be granted, if it might affect the outcome of the suit such that a finding of that fact is necessary and relevant to the proceeding. Fed.Rules Civ.Proc.Rule 56(c), 28 U.S.C.A.

[6] Patents ⚖72(1)
291k72(1) Most Cited Cases

A patent is invalid for anticipation when the same device or method, having all of the elements contained in the claim limitations, is described in a single prior art reference.

[7] Patents ⚖72(1)
291k72(1) Most Cited Cases

To render a patent invalid for anticipation, an anticipating reference must describe the patented subject matter with sufficient clarity and detail to establish that the subject matter existed in the prior art and that such existence would be recognized by persons of ordinary skill in the field of the invention.

[8] Patents ⚖16(2)
291k16(2) Most Cited Cases

[8] Patents ⚖16(3)
291k16(3) Most Cited Cases

[8] Patents ⚖16.13
291k16.13 Most Cited Cases

[8] Patents ⚖36.1(1)
291k36.1(1) Most Cited Cases

Obviousness of device or method claimed in a patent is a legal conclusion based on underlying facts of four general types, all of which must be considered by the trier of fact: (1) the scope and content of the prior art, (2) the level of ordinary skill in the art, (3) the differences between the claimed invention and the prior art, and (4) any objective indicia of nonobviousness.

[9] Patents ⚖16(4)
291k16(4) Most Cited Cases

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

[9] Patents ⚿26(1)
291k26(1) Most Cited Cases

A determination of obviousness of method of device for which patent protection is sought cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention; rather, there must be a teaching or suggestion within the prior art, within the nature of the problem to be solved, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources, to select particular elements, and to combine them as combined by the inventor.

[10] Patents ⚿99
291k99 Most Cited Cases

Inquiry into whether patent is invalid based on lack of a written description is a factual one and must be assessed on a case-by-case basis. 35 U.S.C.A. § 112.

[11] Patents ⚿99
291k99 Most Cited Cases

In order to satisfy the written description requirement, the patent disclosure as originally filed does not have to provide in haec verba support for the claimed subject matter at issue; nonetheless, the disclosure must convey with reasonable clarity to those skilled in the art that the inventor was in possession of the invention. 35 U.S.C.A. § 112.

[12] Patents ⚿99
291k99 Most Cited Cases

Satisfaction of possession test, standing alone, is not always sufficient to meet the written description requirement for patent protection. 35 U.S.C.A. § 112.

[13] Patents ⚿99
291k99 Most Cited Cases

Written description requirement for a patent is satisfied by the patentee's disclosure of such descriptive means as words, structures, figures, diagrams, formulas, etc., that fully set forth the claimed invention; put another way, one skilled in the art, reading the original disclosure, must reasonably discern the limitation at issue in the claims. 35 U.S.C.A. § 112.

[14] Patents ⚿314(5)
291k314(5) Most Cited Cases

Whether a patent claim is enabled is a question of law, although based upon underlying factual findings. 35 U.S.C.A. § 112.

[15] Patents ⚿66(1.25)
291k66(1.25) Most Cited Cases

Two percent limitation for visible reflectance contribution that was claimed in patent for solar control film used in safety and solar control glass was not inherent in, and thus was not anticipated by, existing patent, which disclosed other limitations claimed in patent, but did not claim two percent limitation.

[16] Patents ⚿65
291k65 Most Cited Cases

Inherency of a disclosure in prior art, as will permit a subsequent patent to be rendered invalid due to anticipation, may not be established by probabilities or possibilities, and the mere fact that a certain thing may result from a given set of circumstances is not sufficient.

[17] Federal Civil Procedure ⚿2544
170Ak2544 Most Cited Cases
Party moving for summary judgment has the burden to show that there is an absence of evidence to support the non-moving party's case, and the non-moving party must affirmatively demonstrate by specific factual allegations that a genuine issue of material fact exists for trial. Fed.Rules Civ.Proc.Rule 56(c), 28 U.S.C.A.

[18] Patents ⚿112.5
291k112.5 Most Cited Cases

A patent enjoys a presumption of validity, which can be overcome only through clear and convincing evidence. 35 U.S.C.A. § 282.

[19] Patents ⚿16.14
291k16.14 Most Cited Cases

Patent for solar control film used in safety and solar control glass, which described a film which contributed no more than about two percent visible reflectance, was not invalid for obviousness; no showing was made that prior art contained a teaching, suggestion, or motivation to reduce the reflectance contribution of the solar control film in question.

[20] Patents ⚿99
291k99 Most Cited Cases

Written description and enablement requirements for patents, while related and springing from the same factual predicates, each carry a separate purpose. 35 U.S.C.A. § 112.

[21] Patents ⚿99
291k99 Most Cited Cases

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

Purpose of the enablement requirement for patents is to ensure that the public knowledge is enriched by the patent specification to a degree at least commensurate with the scope of the claims. 35 U.S.C.A. § 112.

[22] Patents ⚖99
291k99 Most Cited Cases

A patent specification may contain a disclosure that is sufficient to enable one skilled in the art to make and use the invention, and yet fail to comply with the description of the invention requirement. 35 U.S.C.A. § 112.

[23] Federal Civil Procedure ⚖2508
170Ak2508 Most Cited Cases

Genuine issue of material fact as to whether a person of ordinary skill in the pertinent art could make or use invention claimed in patent for method of eliminating optical distortion in safety and solar control glass, as would allow patent to satisfy enablement requirement, precluded summary judgment in competitor's suit seeking declaratory judgment regarding validity of patent. 35 U.S.C.A. § 112; Fed.Rules Civ.Proc.Rule 56(c), 28 U.S.C.A.

Patents ⚖328(2)
291k328(2) Most Cited Cases

4,017,661, 5,091,258. Cited.

Patents ⚖328(2)
291k328(2) Most Cited Cases

4,973,511. Valid.
***1370** Joseph T. Leone, DeWitt Ross and Stevens, S.C., of Madison, WI, argued for plaintiffs-appellants. With him on the brief was Joseph A. Ranney.

Gregory E. Upchurch, Thompson Coburn LLP, of St. Louis, MO, argued for defendant-appellee. With him on the brief were Kenneth R. Heineman, and Dudley W. Von Holt.

Before LOURIE, CLEVENGER, and GAJARSA, Circuit Judges.

GAJARSA, Circuit Judge.

Crown Operations International, Ltd., and Mr. Marshall H. Krone (collectively "Crown"), appeal the decision of the United States District Court for the Western District of Wisconsin denying Crown declaratory relief that Solutia's U.S. Patent No. 4,973,511 ("the '511 patent") is invalid for lack of novelty and non-obviousness, and that Solutia's U.S. Patent No. 5,091,258 ("the '258 patent") is invalid for lack of enablement and written description. *Crown Operations Int'l, Ltd. v. Solutia, Inc.*, No. 99-C-802-S, slip op. at 8 (W.D.Wis. Aug. 30, 2000) (memorandum decision and order granting summary judgment) ("*August 30 Order* "); *Crown Operations Int'l, Ltd. v. Solutia, Inc.*, No. 99-C-802-S, slip op. at 24, 27 (W.D.Wis. Aug. 22, 2000) (same) ("*August 22 Order* "). Because we find no error in the district court's opinion with respect to the '511 patent, we affirm that portion of the district court's decision. However, because the district court erred in its analysis of enablement for the '258 patent, and did not address the written description issue for the '258 patent, we vacate the district court's grant of summary judgment on that issue and remand for additional proceedings consistent with this opinion.

I. BACKGROUND

The patents at issue in this appeal relate to layered films used to create safety and solar control glass. An example is an automobile windshield. Most windshields have two layers of glass with a multi-layer film between the glass layers. The multi-layer film adds properties to the glass assembly, such as impact resistance or providing a conductive layer that facilitates defrosting the windshield. An inner layer of the film has solar control properties to selectively reflect, absorb (and thus convert to heat) or transmit defined percentages of certain wavelengths of light. This inner layer is called the solar control film. It is made of a substrate coated by one or more layers of metal or metallic substances. '511 patent, col. 3, l. 64 to col. 4, l. 2. Typically, manufacturers laminate the solar control film between layers of plasticized polyvinyl butyral ("PVB") (sometimes called the "safety film") in a process known as encapsulation. Then, the encapsulated solar control film is sandwiched between two pieces of glass for a final assembly of multi-layer glass with safety and solar control properties.

A. The '511 Patent

The '511 patent is directed to the problem that the metal-coated substrate, *i.e.*, solar control film, tends to wrinkle during encapsulation causing visual distortions. The '511 patent claims to mask the wrinkles from detection by the human eye by ***1371** limiting to two percent or less the visible light reflection contribution of the solar control film compared to reflection from a complete assembly of glass, PVB and solar control film. '511 patent, col. 4, ll. 46-49, col. 8, l. 66 to col. 9, l. 6, col. 14, l. 67 to col. 15, l. 2. Figure 1 from the '511 patent, set forth below, shows the layers in a complete assembly.

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

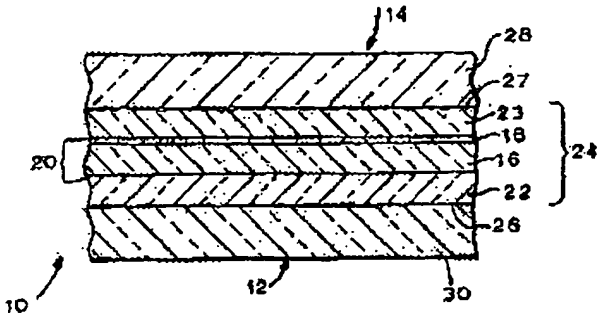


FIG. 1
FIG. 1

The complete safety and solar control glass assembly 10 includes two outer glass layers 28 & 30, PVB layers 22 & 23, and the solar control film 20. The solar control film is comprised of a substrate layer 16 and solar control coating 18. '511 patent, col. 3, ll. 41-53, col. 7, ll. 2-4, col. 10, l. 15. Figure 3 from the '511 patent, set forth below, shows the sub-layers of the solar control coating 18.

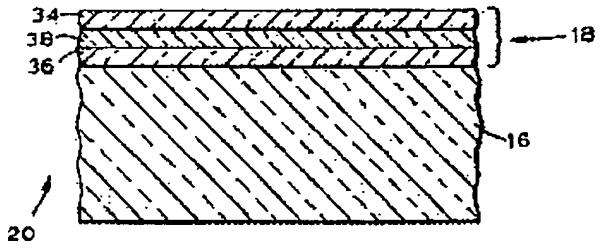


FIG. 3
FIG. 3

Layer 18 is made of multiple sub-layers. Layers 34 and 36 are metal oxide, and layer 38 is metal. '511 patent, col. 5, ll. 12-14. In addition, the ' 511 patent notes that "[p]rior automotive windshields have visible light reflection contributions for their solar films of three percent or greater." Further, it relates that the primary method of achieving a low solar control film reflectance contribution is by providing a specially-designed solar coating. '511 patent, col. 4, ll. 56-65.

On December 16, 1999, Crown sued Solutia (the "Initial Complaint"), seeking, among various other relief, a declaration that the '511 patent was invalid for anticipation and obviousness. Upon the parties' cross-motions for summary judgment, the district court found the '511 patent not anticipated and not invalid for obviousness. *August 22 Order* at 24, 27. We discuss herein only those portions of the *August 22 Order* relevant to the issues on appeal, which relate solely to the summary judgment finding that the '511 patent was not ***1372** invalid on the grounds of anticipation and obviousness.

Claim 1, the only independent claim of the '511 patent, is set forth below, with the element numbers from Figure 1 inserted into the claim.

> 1. A composite solar/safety film **[24]** for use in a laminated window assembly **[10]** comprising:
> a flexible, transparent plastic substrate layer **[16]** having a carrier surface and an opposing back surface;
> a multilayer solar control coating **[18]** on said carrier surface, said coated substrate defining a solar control film **[20]**; and
> at least one flexible, transparent, energy absorbing plastic safety layer **[23 and/or 22]** bonded to a surface of said solar control film;
> wherein said ***solar control film contributes no more than about 2% visible reflectance,*** based on total visible incident radiation, in a laminated window assembly containing said composite solar/safety film laminated to at least one rigid transparent member **[30 and/or 28].**

'511 patent, col. 14, l. 57 to col. 15, l. 4 (emphasis added and emphasized numbers added to identify elements shown in Figure 1 above).

Crown argued that U.S. Patent No. 4,017,661 to Gillery (the "Gillery patent") anticipates the '511 patent. The district court held otherwise, because, while the Gillery patent discloses the first three limitations of claim 1 of the '511 patent, it does not disclose the two percent visible reflectance limitation. The court found that neither the Gillery patent claims nor its description expressly disclose a two percent limit on reflectance contribution from the solar control film layer. Crown argued that the two percent limitation was inherently present in the Gillery patent's teachings because the Gillery patent disclosed an assembly with PVB layers, substrate layer, and substrate metal-coating--arguably of the same composition and thickness of the films disclosed by the '511 patent. Thus, Crown argued, because the structure, thickness and materials of the assembly were the same or within the same range(s), the Gillery patent must inherently disclose a two percent limitation. The district court rejected this argument because it found that none of the embodiments disclosed by the Gillery patent meet the two percent visible light reflectance limit. [FN1]

> FN1. The district court, applying a similar analysis, also found that UK Patent Application GB 2 057 355 (the "UK patent") did not anticipate the '511 patent because it did not have the two percent limitation.

In its *August 22 Order,* the district court also held that the '511 patent was not rendered invalid for obviousness by Gillery or the other prior art cited by Crown because no

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

prior art discloses: (i) that reflectance below two percent will mask wrinkles; (ii) a solar control film layer with reflectance below two percent; or (iii) any suggestion, motivation or teaching to reduce solar control film visible light reflectivity below two percent. Although the prior art generally sought to reduce visible light reflectivity, it also taught disadvantages of a very thin metal-coating on the substrate, including sacrificing infrared reflectivity. Thus, it taught that the proper compromise to achieve the conflicting goals of infrared (non-visible light) reflectance, visible light transmission and conductivity ***1373** was a solar control film with a visible light reflectivity greater than two percent.

B. The '258 Patent

The '258 patent is directed at eliminating optical distortion, called "applesauce," in safety and solar control glass assemblies of the type discussed above for the '511 patent. The '258 patent discloses a method to control distortion otherwise caused by the safety and solar film layer by measuring and controlling the texture of the surface of the PVB layers. The method expresses texture using a "wave index" and a "roughness value." The wave index calculation is at issue in this appeal. Wave index indicates the relative waviness of the surface of the PVB. Determining wave index involves measuring the surface of the PVB and then aggregating the measurements into a single number, the wave index, through a calculation purportedly described in the '258 patent.

The '258 patent directs one to use an instrument to physically measure the waviness of the surface of the PVB and capture the measurement into an electronic "trace line" representing the contours of the PVB surface. '258 patent, col. 7, II. 54-65. Since the "trace line" is stored electronically, a computer program is used to calculate wave index from the trace. Three figures from the '258 patent, given below, provide examples of PVB surface trace lines.



FIG. 7



FIG. 8

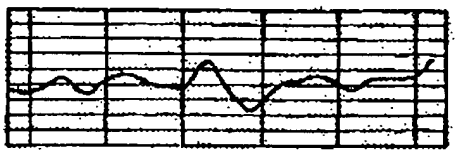


FIG. 9

The rules for calculating the wave index implement a "smoothing" function. The smoothing process seeks to eliminate minor inflection points (peaks or valleys) to simplify the calculation of wave index. '258 patent, col. 7, I. 66 to col. 8, I. 2.

In the Initial Complaint, Crown sought a declaration that the '258 patent was invalid for anticipation and obviousness. Then, on May 26, 2000, Crown amended the complaint (the "Amended Complaint") to additionally claim in Count VI that the '258 patent is invalid under 35 U.S.C. § 112, first paragraph, because it lacked enablement and written description due to ambiguities in the disclosed wave index calculation. In its *August 22 Order,* the district court found the '258 patent not anticipated and not invalid for obviousness. *August 22 Order* at 28-29.

With respect to Count VI of Crown's amended complaint, Solutia moved for ***1374** summary judgment on Crown's enablement and written description claim. Crown opposed Solutia's summary judgment motion, arguing that the '258 patent did not meet the enablement and written description requirements. The district court found the '258 patent not invalid for lack of enablement, but did not discuss in its opinion the written description requirement. *August 30 Order* at 8-13. We discuss herein only those portions of the *August 30 Order* relevant to the issues on appeal, which relate to summary judgment finding the '258 patent not invalid on the grounds of enablement and the procedural disposition of the written description issue.

Claim 1 of the '258 patent is set forth below. In the language of this claim, "laminate" refers to the complete glass, PVB and solar control film assembly, and "functional performance layer" refers to the solar control coating. '258 patent, col. 3, II. 45-65.

> 1. A laminate which is substantially free of reflected distortion when used in a safety glazing comprising:
> a transparent, thermoplastic substrate layer, optionally surface treated or coated, bearing one or more functional performance layers; and
> at least one layer of plasticized polyvinyl butyral bonded on one side to a functional performance layer or the substrate layer and having a roughened deairing surface on its other side characterized by a roughness value, Rz, of at least 10 micrometers;
> said at least one plasticized polyvinyl butyral [PVB] layer, before bonding to the substrate layer or functional performance layer, ***possessing low surface waviness on each side characterized by a wave index value, WI, of less than 15,000 square micrometers.***

'258 patent, col. 12, II. 2-16 (emphasis added).

Crown argued that the rules disclosed by the '258 patent for calculating wave index are not sufficiently precise to enable

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

a person of ordinary skill in the art to practice the '258 patent without undue experimentation. The wave index calculation as described by the '258 patent is set forth below.

> In this regard, considering the waviness profile as a series of peaks and valleys, the smoothing rules of the program consider an inflection point to be a true peak or valley if it is: i) at least 100 micrometers away from the immediately preceding prior peak or valley and ii) at least 0.5 micrometer above or below the immediately preceding prior peak or valley, a valley being at least 0.5 micrometer below the immediately preceding prior peak. Pitch (P) is the distance between one valley and the next valley or in other words across the base of a peak. Average amplitude (H avg) and average pitch (P avg) are determined by the program for the smoothed trace of ten 12.5 mm tracing lengths (the second five lengths being 90 o to the first five lengths). From the average of the averaged H's and P's, a WI value is computed from the equation: Wave Index (WI) = (H avg) x (P avg) where H avg and P avg are in microns.

'258 patent, col. 8, ll. 3-19.

Crown asserted that according to the disclosed wave index "calculation," one of ordinary skill in the pertinent art would not know whether to instruct the smoothing program to disregard a peak by comparing it to an immediately preceding peak, or to a valley. The district court held that common sense and the clarifying clause "a valley being at least 0.5 micrometer ***1375** below the immediately preceding prior peak" defeated Crown's argument. Thus, the district court held that the alleged grammatical ambiguities in the rules disclosed for calculating wave index did not invalidate the patent for lack of enablement.

Crown timely appealed the district court's two orders, raising the issues of anticipation and obviousness of the '511 patent, and lack of enablement and written description of the '258 patent. We have jurisdiction pursuant to 28 U.S.C. § 1295(a)(1).

II. STANDARD OF REVIEW

[1][2][3][4][5] We review a district court's grant of summary judgment without deference. *Atmel Corp. v. Info. Storage Devices, Inc.*, 198 F.3d 1374, 1378, 53 USPQ2d 1225, 1227 (Fed.Cir.1999). Summary judgment is appropriate when the moving party demonstrates that "there is no genuine issue as to any material fact and that the moving party is entitled to a judgment as a matter of law." Fed.R.Civ.P. 56(c); *Celotex Corp. v. Catrett*, 477 U.S. 317, 322-23, 106 S.Ct. 2548, 91 L.Ed.2d 265 (1986). On summary judgment, the evidence must be viewed in the light most favorable to the party opposing the motion, *Poller v. Columbia Broad. Sys., Inc.*, 368 U.S. 464, 473, 82 S.Ct. 486, 7 L.Ed.2d 458 (1962), with doubts resolved in favor of the nonmovant, *Cantor v. Detroit Edison Co.*, 428 U.S. 579, 582, 96 S.Ct. 3110, 49 L.Ed.2d 1141 (1976); *Transmatic, Inc. v. Gulton Indus., Inc.*, 53 F.3d 1270, 1274, 35 USPQ2d 1035, 1038 (Fed.Cir.1995). Once the moving party has satisfied its initial burden, the opposing party must establish a genuine issue of material fact and cannot rest on mere allegations, but must present actual evidence. *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248, 106 S.Ct. 2505, 91 L.Ed.2d 202 (1986). Issues of fact are genuine only "if the evidence is such that a reasonable jury could return a verdict for the nonmoving party." *Id.* A disputed fact is material if it might affect the outcome of the suit such that a finding of that fact is necessary and relevant to the proceeding. *Id.; General Mills, Inc. v. Hunt-Wesson, Inc.*, 103 F.3d 978, 980, 41 USPQ2d 1440, 1442 (Fed.Cir.1997).

[6][7] A patent is invalid for anticipation when the same device or method, having all of the elements contained in the claim limitations, is described in a single prior art reference. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed.Cir.1989); *Perkin-Elmer Corp. v. Computervision Corp.*, 732 F.2d 888, 894, 221 USPQ 669, 673 (Fed.Cir.1984). An anticipating reference must describe the patented subject matter with sufficient clarity and detail to establish that the subject matter existed in the prior art and that such existence would be recognized by persons of ordinary skill in the field of the invention. *See In re Spada*, 911 F.2d 705, 708, 15 USPQ2d 1655, 1657 (Fed.Cir.1990); *Diversitech Corp. v. Century Steps, Inc.*, 850 F.2d 675, 678, 7 USPQ2d 1315, 1317 (Fed.Cir.1988).

[8] Obviousness is a legal conclusion based on underlying facts of four general types, all of which must be considered by the trier of fact: (1) the scope and content of the prior art; (2) the level of ordinary skill in the art; (3) the differences between the claimed invention and the prior art; and (4) any objective indicia of nonobviousness. *See Graham v. John Deere Co.*, 383 U.S. 1, 17-18, 86 S.Ct. 684, 15 L.Ed.2d 545 (1966); ***1376***Continental Can Co. USA, Inc. v. Monsanto Co.*, 948 F.2d 1264, 1270, 20 USPQ2d 1746, 1750-51 (Fed.Cir.1991); *Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561, 1566-68, 1 USPQ2d 1593, 1594 (Fed.Cir.1987).

[9] "Determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention." *ATD Corp. v. Lydall, Inc.*, 159 F.3d 534, 546, 48 USPQ2d 1321, 1329 (Fed.Cir.1998). There must be a teaching or suggestion within the prior art, within the nature of the problem to be solved, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources, to select particular elements, and to combine them as combined by the

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

inventor. *See Ruiz v. A.B. Chance Co.*, 234 F.3d 654, 665, 57 USPQ2d 1161, 1167 (Fed.Cir.2000); *ATD Corp.*, 159 F.3d at 546, 48 USPQ2d at 1329; *Heidelberger Druckmaschinen AG v. Hantscho Commercial Prods., Inc.*, 21 F.3d 1068, 1072, 30 USPQ2d 1377, 1379 (Fed.Cir.1994) ("When the patented invention is made by combining known components to achieve a new system, the prior art must provide a suggestion or motivation to make such a combination.").

[10][11][12][13] The written description inquiry is a factual one and must be assessed on a case-by-case basis. *See Vas-Cath Inc. v. Mahurkar*, 935 F.2d 1555, 1561, 19 USPQ2d 1111, 1116 (Fed.Cir.1991) (quoting *In re Smith*, 59 C.C.P.A. 1025, 458 F.2d 1389, 1395, 173 USPQ 679, 683 (CCPA 1972) ("Precisely how close the original description must come to comply with the description requirement of § 112 must be determined on a case-by-case basis."). In order to satisfy the written description requirement, the disclosure as originally filed does not have to provide in haec verba support for the claimed subject matter at issue. *See Fujikawa v. Wattanasin*, 93 F.3d 1559, 1570, 39 USPQ2d 1895, 1904 (Fed.Cir.1996). Nonetheless, the disclosure must convey with reasonable clarity to those skilled in the art that the inventor was in possession of the invention, *Vas-Cath Inc.*, 935 F.2d at 1563-64, 19 USPQ2d at 1116-17, although we have also clarified that the possession test alone is not always sufficient to meet the written description requirement, *Enzo Biochem, Inc. v. Gen-Probe Inc.*, 285 F.3d 1013, 1020-21 (Fed.Cir.2002). As such, "the written description requirement is satisfied by the patentee's disclosure of 'such descriptive means as words, structures, figures, diagrams, formulas, etc., that fully set forth the claimed invention.' " *Enzo Biochem*, 285 F.3d at 1021 (quoting *Lockwood v. American Airlines, Inc.*, 107 F.3d 1565, 1572, 41 USPQ2d 1961, 1966 (Fed.Cir.1997)). Put another way, one skilled in the art, reading the original disclosure, must reasonably discern the limitation at issue in the claims. *Waldemar Link, GmbH & Co. v. Osteonics Corp.*, 32 F.3d 556, 558, 31 USPQ2d 1855, 1857 (Fed.Cir.1994).

[14] Whether a claim is enabled under 35 U.S.C. § 112, first paragraph is a question of law, although based upon underlying factual findings. See *PPG Indus., Inc. v. Guardian Indus. Corp.*, 75 F.3d 1558, 1564, 37 USPQ2d 1618, 1623 (Fed.Cir.1996); *In re Goodman*, 11 F.3d 1046, 1049-50, 29 USPQ2d 2010, 2013 (Fed.Cir.1993).

III. DISCUSSION

A. The '511 Patent

On appeal, Crown describes various purported errors in the district court's analysis ***1377** of the validity of the '511 patent. Despite Crown's contentions, we ascertain no error requiring reversal of the district court's determination of validity over Crown's claims of anticipation and obviousness.

[15][16] Regarding alleged anticipation by the Gillery patent, on its face the Gillery patent does not disclose or discuss a two percent limitation for the reflectance contribution of the solar control film. Crown maintains that the '511 patent merely claims a preexisting property inherent in the structure disclosed in the prior art. Crown urges us to accept the proposition that if a prior art reference discloses the same structure as claimed by a patent, the resulting property, in this case, two percent solar control film reflectance, should be assumed. We decline to adopt this approach because this proposition is not in accordance with our cases on inherency. If the two percent reflectance limitation is inherently disclosed by the Gillery patent, [FN2] it must be necessarily present and a person of ordinary skill in the art would recognize its presence. *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed.Cir.1999); *Continental Can*, 948 F.2d at 1268, 20 USPQ2d at 1749. Inherency "may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." *Id.* at 1269, 20 USPQ2d at 1749 (quoting *In re Oelrich*, 666 F.2d 578, 581, 212 USPQ 323, 326 (CCPA 1981)).

> FN2. In order to claim "equivalent structure" between the Gillery patent and the '511 patent, Crown's inherency argument rests on a precondition of its own making--that the Gillery patent discloses use of TiO2, even though it specifies TiOx, where x is greater than 1.0 but less than 2.0. Although Crown vigorously argues this point, we do not reach this issue because even if Crown is correct that the structures are equivalent, Crown's inherency argument fails for the reasons set forth herein.

[17][18] In arguing inherent disclosure of the two percent limitation in the Gillery patent, Crown bears an evidentiary burden to establish that the limitation was necessarily present. [FN3] The moving party in a summary judgment motion has the burden to show "that there is an absence of evidence to support the non-moving party's case;" the non-moving party must affirmatively demonstrate by specific factual allegations that a genuine issue of material fact exists for trial. *Celotex Corp. v. Catrett*, 477 U.S. 317, 322-23, 106 S.Ct. 2548, 91 L.Ed.2d 265 (1986). A patent enjoys a presumption of validity, *see* 35 U.S.C. § 282, which can be overcome only through clear and convincing evidence, *see United States Surgical Corp. v. Ethicon, Inc.*, 103 F.3d 1554, 1563, 41 USPQ2d 1225, 1232 (Fed.Cir.1997). Given the presumption of validity afforded

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

the '511 patent, Crown has failed to meet its burden because it has not presented sufficient evidence to rebut the facial evidence offered by Solutia that the Gillery patent does not ***1378** disclose the two percent limitation. *See Eli Lilly & Co. v. Barr Lab. Inc.*, 251 F.3d 955, 962, 58 USPQ2d 1869, 1874 (Fed.Cir.2001) ("[A] moving party seeking to have a patent held not invalid at summary judgment must show that the nonmoving party, who bears the burden of proof at trial, failed to produce clear and convincing evidence on an essential element of a defense upon which a reasonable jury could invalidate the patent."); *In re Robertson*, 169 F.3d at 745 (recognizing that extrinsic evidence may be required to establish inherency). Instead, Crown offers only an assumption and its own contentions. [FN4]

> FN3. Crown's reliance on *Pall Corp. v. Micron Separations, Inc.*, 66 F.3d 1211, 36 USPQ2d 1225 (Fed.Cir.1995), and *O.I. Corp. v. Tekmar Co.*, 115 F.3d 1576, 42 USPQ2d 1777 (Fed.Cir.1997), to characterize the two percent limitation as a "performance limitation" similar to the claim terms at issue in those cases is unpersuasive and overbroad. Respectively, *Pall* and *Tekmar* dealt with the claim terms "skinless" and "passage." Beyond the readily apparent difference between these potentially broad terms and the precise specification of a two percent limit in the '511 patent, characterizing a claim limitation as a "performance characteristic" is not helpful as to whether the "necessarily present" requirement of inherency is met.
>
> FN4. As indicated by this Court's questions at oral argument concerning the seemingly direct route to prove that the Gillery patent contains the two percent limitation implementing an embodiment of the Gillery patent and testing it this Court finds puzzling Crown's reluctance regarding this approach to generate extrinsic proof that the Gillery patent inherently meets the two percent limitation.

Crown also argues that the district court erred by comparing reflectance values in the Gillery patent to non-corresponding values in the '511 patent. *August 22 Order* at 23-24. While perhaps the district court could have been more careful to explain the basis of its comparison, on a close reading of the district court's analysis we find that the alleged improper comparison only supported the district court's primary point that no embodiment of the Gillery patent disclosed the two percent limitation, a conclusion that Crown has not shown to be in error.

[19] Finally, Crown argues that various prior art references invalidate the '511 patent as obvious in view of such prior art. Crown's arguments lack merit because it has not shown that the prior art contains a teaching, suggestion or motivation to reduce the reflectance contribution of the solar control film to "no more than about two percent," and the district court properly concluded that there was no such teaching, suggestion or motivation in the prior art cited by Crown. *See Ruiz*, 234 F.3d at 665, 57 USPQ2d at 1167; *In re Rouffet*, 149 F.3d 1350, 1359, 47 USPQ2d 1453, 1459 (Fed.Cir.1998).

B. The '258 Patent

On appeal, Crown argues that the district court erred in analyzing the impact of the ambiguities in the wave index calculation on the enablement requirement for the '258 patent. In addition to its enablement attack, Crown also argues that the '258 patent does not meet the written description requirement of § 112, first paragraph.

[20][21][22] The two requirements, while related and springing from the same factual predicates, [FN5] each carry a separate purpose. The purpose of the enablement requirement is to "ensure[] that the public ***1379** knowledge is enriched by the patent specification to a degree at least commensurate with the scope of the claims." *Nat'l Recovery Techs., Inc. v. Magnetic Separation Sys.*, 166 F.3d 1190, 1196, 49 USPQ2d 1671, 1675 (Fed.Cir.1999). One of our predecessor courts has held the enablement and written description requirements to be separate and distinct, and has held that a "specification may contain a disclosure that is sufficient to enable one skilled in the art to make and use the invention and yet fail to comply with the description of the invention requirement." *In re Barker and Pehl*, 559 F.2d 588, 591, 194 USPQ 470, 472 (CCPA 1977). Subsequently, this court has held that the purpose of the written description is distinct from merely explaining how to make and use the invention. *See Enzo Biochem*, 285 F.3d at 1020-22; *Vas-Cath*, 935 F.2d at 1563-64, 19 USPQ2d at 1117. In light of the odd procedural setting of the written description issue in this appeal, our disposition of this appeal based on enablement, and given that the two requirements are distinct and each are necessary, we do not reach the written description issue except to note that it appears to remain available for adjudication or disposition by the district court on remand. [FN6]

> FN5. Also springing from these same underlying factual predicates is the § 112, second paragraph, definiteness requirement. This requirement is distinct from the enablement and description requirements, which arise from § 112, first paragraph.
> [D]efiniteness and enablement are analytically distinct requirements, even though both concepts are contained in 35 U.S.C. § 112. The definiteness

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

> requirement of 35 U.S.C. § 112, ¶ 2 is a legal requirement, based on the court's role as construer of patent claims ... Definiteness requires the language of the claim to set forth clearly the domain over which the applicant seeks exclusive rights.... The test for whether a claim meets the definiteness requirement is "whether one skilled in the art would understand the bounds of the claim when read in light of the specification."
> *Process Control Corp.*, 190 F.3d at 1358 n. 2, 52 USPQ2d at 1034 n. 2 (internal citations omitted). *See also* 3 Donald S. Chisum, Chisum on Patents, § 8.03 at 8-14 (2001) (noting the difference between the requirements of "definiteness, which claims must meet, from the requirements of enablement, which the disclosures of the specification must meet").
>
> FN6. Based on the record before us, the written description issue has the following procedural posture: (i) Crown's Count VI of its amended complaint raised the written description issue; (ii) Solutia's summary judgment motion argued that the '258 patent met the written description requirement; (iii) in opposition Crown argued that the written description requirement was not met; (iv) the district court did not dispose of the written description issue or discuss the issue in its opinion in a way that enables our review; and (v) Crown preserved the written description issue in its appeal to this court and thus has not waived its further adjudication on remand.

[23] Turning to the enablement issue, we agree with Crown that the ambiguities and lack of specified boundary conditions, and Crown's proffered evidence concerning the same, raise a genuine issue of material fact as to whether a person of ordinary skill in the pertinent art could make or use the invention of the '258 patent [FN7] without undue experimentation. *White Consol. Indus. v. Vega Servo-Control,* 713 F.2d 788, 791, 218 USPQ 961, 963-64 (Fed.Cir.1983). The district court found otherwise. However, it appears not to have considered the statements of Crown's expert concerning the effect of unspecified boundary conditions on the calculation of wave index.

> FN7. All seventeen claims of the '258 patent refer to wave index, thus they all stand or fall together.

Following the reasoning of the district court, Solutia argues that a person of ordinary skill in the pertinent art could overcome any ambiguities in the wave index calculation without undue experimentation by testing a limited number of possibilities for computing the wave index. In response, Crown offers statements of its expert that the '258 patent does not define amplitude and that a person of ordinary skill in the art would not know whether to measure amplitude: (i) from a centerline running horizontally through the "middle" of the trace; (ii) from "peak-to-peak," *i.e.,* from the bottom of a valley to the top of a peak; or (iii) from some other baseline or reference running horizontally somewhere through the trace. On its face, the '258 patent does not define amplitude. However, average amplitude directly impacts the wave index calculation because wave index ***1380** is the result of multiplying average amplitude by average pitch. Simply put, the wave index calculation would produce two separate numbers if calculated with a centerline versus a "peak-to-peak" amplitude. Worse yet, a range of various wave index values are possible for amplitude baselines running horizontally somewhere through the trace at various locations. To show that the wave index calculation is enabled, Solutia cites various details from the '258 patent concerning how to perform the test to generate a trace of the PVB surface to calculate wave index. However, Solutia does not present sufficient evidence to rebut Crown's demonstration of the amplitude ambiguity in the wave index calculation. This is so because: (i) the amplitude is a direct input to the critical claim limitation, a wave index of less than 15,000 square micrometers; and (ii) the novel aspects of the invention must be disclosed and not left to inference, that is, a patentee may not rely on the inference of a person of ordinary skill in the pertinent art to supply such novel aspects. *See Genentech Inc. v. Novo Nordisk A/S,* 108 F.3d 1361, 1366, 42 USPQ2d 1001, 1005 (Fed.Cir.1997) (stating that the knowledge of a hypothetical person of ordinary skill in the art cannot be used to supply the patentable aspects of the invention).

Compounding the amplitude ambiguity, Crown also notes that the wave index is the result of two independently varying, unbounded terms: average pitch and average amplitude. On its face, this does not seem to be a problem. However, Crown's expert noted that because boundary conditions are not specified, the claim covers inoperative embodiments. For example, a wave index of 15,000 square micrometers results from an average height of 1000 micrometers multiplied by an average pitch of 15 micrometers. Yet, according to Crown's expert, an average height of 1000 micrometers would not be acceptable for the PVB. As with the amplitude ambiguity, the problem goes well beyond this single example because a full range of resulting inoperative embodiments are possible for values of average height and average pitch that, when multiplied, produce a wave index value that meets the limitation of the claim. Such inoperative embodiments do not necessarily invalidate the claim. *See Atlas Powder Co. v. E.I. du Pont De Nemours & Co.,* 750 F.2d 1569, 1576-77, 224 USPQ 409, 414 (Fed.Cir.1984); *In re Cook,* 58 C.C.P.A. 1049, 439 F.2d 730, 735, 169 USPQ 298, 302 (1971) (noting that although claims may read on some inoperative

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

embodiments, this does not necessarily invalidate the claim if the necessary information to limit the claims to operative embodiments is known to a person of ordinary skill in the art). [FN8] However, the inoperative embodiments support Crown's assertion that there is a genuine issue of material fact with respect to enablement. *See Atlas Powder,* 750 F.2d at 1576-77; *see also Process Control Corp. v. HydReclaim Corp.,* 190 F.3d 1350, 1358-59, 52 USPQ2d 1029, 1034-35 (Fed.Cir.1999) (holding that the district court failed in its ***1381** claim construction to consider the effect of inoperative embodiments on invalidity due to lack of enablement). [FN9]

FN8. The court in *In re Cook* further notes that a claim may be invalid if it reads on significant numbers of inoperative embodiments. *In re Cook,* 439 F.2d at 734, 169 USPQ at 301-02 (citing *Graver Tank & Mfg. Co. v. Linde Air Products Co.,* 336 U.S. 271, 276-77, 69 S.Ct. 535, 93 L.Ed. 672, 80 USPQ 451, 453 (1949)). *See also In re Moore,* 439 F.2d 1232, 1236, 169 USPQ 236, 239 (CCPA 1971) (noting that the question is whether the scope of enablement conveyed by the disclosure to a person of ordinary skill in the art is commensurate with the scope of protection taught by the claims); Chisum, § 7.03[7][a] at 7-108 & n. 6.

FN9. The inoperative embodiment inquiry informs the enablement inquiry; they are not the same inquiry. *Nat'l Recovery Techs.,* 166 F.3d at 1196, 49 USPQ2d at 1676.

Further compounding the ambiguities with the wave index rules, the ' 258 patent's rules for determining which inflection points are "true" inflection points additionally support Crown's argument that it has raised a genuine issue of material fact. Crown demonstrated in various ways through its experts and arguments the potential indeterminacy in the rules. Solutia's expert admitted that there was some ambiguity in the rules with respect to whether a preceding peak or valley was the reference point in selecting a "true" peak or valley.

Solutia argues that even if the disclosed wave index calculation has ambiguities and is indeterminate, a person of ordinary skill in the pertinent art would be able to make and use the invention with some experimentation, but less than "undue" experimentation. Solutia argues that such a skilled person would only have to try two possibilities for amplitude, centerline and "peak- to-peak," and that experimenting to discover which of two possibilities to use is well within the boundary of undue experimentation. Crown counters that the amplitude ambiguity and potential inoperative embodiments, combined with the ambiguities in the smoothing rules, seems to suggest a wide range of possibilities which one must try. [FN10] With this wide range of possibilities, we agree that Crown has raised a genuine issue of material fact as to the amount and type of experimentation required, facts that will determine whether such experimentation is undue. *See Enzo Biochem Inc., v. Calgene Inc.,* 188 F.3d 1362, 1371, 52 USPQ2d 1129, 1135-36 (Fed.Cir.1999) (holding that a reasonable amount of experimentation does not invalidate a patent, but undue experimentation does invalidate, and holding that the *Wands* factors, which determine whether a patent's disclosure is insufficient such that the experimentation required would be undue, apply to inter partes litigation). [FN11] While ultimately a trier of fact may reach the conclusion that any required experimentation is not undue, Crown has shown that sufficient potential for undue experimentation exists such that disposal on summary judgment is improper.

FN10. We note that the specification for the '258 patent states that in the disclosed embodiment the wave index is calculated using a software program running on a personal computer being fed the trace line. '258 patent, col. 7, ll. 64-68. Undoubtedly, Solutia took care to ensure that the program contained the necessary boundary conditions and other information to calculate wave index to practice the invention. It appears, however, that Solutia took substantially less care in transcribing the information from the program into the specification's rules for calculating wave index. This incongruity will be relevant to the question of enablement upon remand. *See* Chisum, § 7.03[4][e] at 7-86 & n. 77 ("A specification that claims an invention requiring implementation through computer software but fails to set forth the details of computer programming may present issues of whether the experimentation required to write the programming is reasonable or unreasonable.") (summarizing the teachings of various cases).

FN11. The *Wands* factors are:
(1) the quantity of experimentation necessary, (2) the amount of direction or guidance presented, (3) the presence or absence of working examples, (4) the nature of the invention, (5) the state of the prior art, (6) the relative skill of those in the art, (7) the predictability or unpredictability of the art, and (8) the breadth of the claims. *In re Wands,* 858 F.2d 731, 737, 8 USPQ2d 1400, 1404 (Fed.Cir.1988).

***1382** CONCLUSION

Because we hold that the '511 patent has not been shown to be invalid due to anticipation or obviousness and that a genuine issue of material fact exists with respect to facts

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

underlying the determination of enablement for the ' 258 patent, we affirm-in-part and reverse-in-part the district court's decision and remand for additional proceedings consistent with this opinion.

AFFIRMED-IN-PART, REVERSED-IN-PART, AND REMANDED.

COSTS

Each party bears its own costs.

289 F.3d 1367, 62 U.S.P.Q.2d 1917

END OF DOCUMENT

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

United States Court of Appeals,
Federal Circuit.

In re Philip R. THRIFT and Charles T. Hemphill.

No. 01-1445.

Aug. 9, 2002.

The Board of Patent Appeals and Interferences affirmed examiner's rejection for obviousness of claims in application for patent on voice activated hypermedia systems using grammatical metadata. Applicants appealed. The Court of Appeals for the Federal Circuit, Dyk, Circuit Judge, held that: (1) claim phrase "speech user agent" would be interpreted, consistent with its plain meaning, as an interface that allows the user to interact with the system by speaking, without limitation to an agent that "dynamically creates the vocabulary, grammar and actions that are possible for the user to use in a given situation;" (2) examiner satisfied his burden of presenting a prima facie case of obviousness with respect to claim 1, based on a combination of two prior art references; but (3) rejection as obvious of claim which added to claim 1 specific grammar-creation capability features, on ground "[t]he use of grammar is old and well known in the art of speech recognition as a means of optimization which is highly desirable," was inadequate.

Affirmed in part and vacated and remanded in part.

West Headnotes

[1] Patents ⇐113(6)
291k113(6) Most Cited Cases

Conclusory argument that the examiner rejected dependent claims without providing any evidence of the additional features of these claims was not sufficient to raise separate issues on appeal with respect to the dependent claims, where applicants failed to identify the specific errors in the decision or even the features of these claims that would allow them to overcome an obviousness rejection, and thus, all of the dependent claims had to stand or fall with their respective independent claims.

[2] Patents ⇐113(6)
291k113(6) Most Cited Cases

The ultimate determination of obviousness by the Board of Patent Appeals and Interferences is reviewed without deference, and the Board's underlying factual findings are reviewed for substantial evidence. 35 U.S.C.A. § 103(a).

[3] Patents ⇐16.5(1)
291k16.5(1) Most Cited Cases

To establish a prima facie case of obviousness, the Board of Patent Appeals and Interferences must, inter alia, show some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references, and the motivation, suggestion, or teaching may come explicitly from statements in the prior art, the knowledge of one of ordinary skill in the art, or, in some cases the nature of the problem to be solved. 35 U.S.C.A. § 103.

[4] Patents ⇐101(3)
291k101(3) Most Cited Cases

In determining whether the examiner satisfied his burden of presenting a prima facie case of obviousness with respect to claim 1 of patent application for voice activated hypermedia systems using grammatical metadata, applicants' limiting interpretation of the claim phrase "speech user agent" as an agent that "dynamically creates the vocabulary, grammar and actions that are possible for the user to use in a given situation" was inappropriate, particularly in light of the grammar-creation capability separately recited in subsequent claims, and, instead, the phrase "speech user agent" would be interpreted consistent with its plain meaning: an interface that allows the user to interact with the system by speaking. 35 U.S.C.A. § 103.

[5] Patents ⇐157(1)
291k157(1) Most Cited Cases

[5] Patents ⇐167(1)
291k167(1) Most Cited Cases

When examining claims for patentability, claims are interpreted as broadly as is reasonable and consistent with the specification.

[6] Patents ⇐113(6)
291k113(6) Most Cited Cases

Reliance by the Board of Patent Appeals and Interferences on "common knowledge and common sense" does not fulfill the agency's obligation to cite references to support its conclusions; instead, the Board must document its reasoning on the record to allow accountability and effective judicial review.

[7] Patents ⇐36(3)
291k36(3) Most Cited Cases

Examiner satisfied his burden of presenting a prima facie case of obviousness with respect to claim 1 of patent application for voice activated hypermedia systems using grammatical metadata, based on a combination of two prior art references, where the examiner clearly identified a

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

motivation to combine the references, stating that the skilled artisan would have "found it obvious to incorporate the speech input and speech recognition techniques taught by [one reference] into the expert system of [the other] in order to reduce the need for less user friendly manual keyboard and mouse click inputs," and the motivation to combine the references was present in the text of each reference. 35 U.S.C.A. § 103.

[8] Patents ⚿108
291k108 Most Cited Cases

On an application for a patent on voice activated hypermedia systems using grammatical metadata, rejection as obvious of claim which added to properly- rejected claim 1 specific grammar-creation capability features, on ground that "[t]he use of grammar is old and well known in the art of speech recognition as a means of optimization which is highly desirable," was inadequate on its face since the examiner's statement, while generally addressing the use of grammar, did not discuss the claim's unique limitations of extracting, modifying, or processing the grammar to interact with hypermedia sources. 35 U.S.C.A. § 103.

[9] Patents ⚿113(6)
291k113(6) Most Cited Cases

Rejection of patent application claim as obvious could not be sustained on appeal from the Board of Patent Appeals and Interferences on theory that the applicants did not properly raise the grounds for sustaining the claim before the Board, where the Board did not hold that those arguments were untimely and did reach the merits, but failed to provide an adequate basis for rejecting the claim.

Patents ⚿328(2)
291k328(2) Most Cited Cases

5,333,237. Cited As Prior Art.
***1359** Ronald O. Neerings, Texas Instruments Incorporated, of Dallas, Texas, argued for appellants. Of counsel on the brief was Jay M. Cantor, Baker & Botts L.L.P., of Washington, DC, Of counsel was Michael K. Skrehot, Texas Instruments Incorporated, of Dallas, TX.

Joseph G. Piccolo, Associate Solicitor, United States Patent and Trademark Office, of Arlington, Virginia, argued for the Director of the United States Patent and Trademark Office. With him on the brief were John M. Whealan, Solicitor, and Cynthia C. Lynch, Associate Solicitor.

Before BRYSON, GAJARSA, and DYK, Circuit Judges.

DYK, Circuit Judge.

Philip R. Thrift and Charles T. Hemphill appeal from the decision of the Board of Patent Appeals and Interferences ("Board") affirming the examiner's rejection of claims 1-19 for obviousness under 35 U.S.C. § 103. *Ex parte Thrift*, No.1998-1109 (Bd. Pat.App. & Int. Feb. 28, 2001) ("*Decision on Appeal* "); *Ex parte Thrift*, No.1998-1109 (Bd. Pat.App. & Int. Apr. 24, 2001) ("*Decision on Request for Rehearing* "). We affirm the Board's decision with respect to claims 1-10, but vacate the Board's decision with respect to claims 11-19, and remand for further proceedings with respect to these claims.

BACKGROUND

On April 10, 1995, Thrift and Hemphill filed U.S. Patent Application No. 08/419,229 (the "'229 application") entitled "Voice Activated HyperMedia Systems Using Grammatical Metadata." On June 6, 1995, Thrift and Hemphill assigned their entire rights in the invention described in the '229 application to Texas Instruments, Inc. ***1360** ("TI"), making TI the real party in interest in this proceeding.

[1] The parties focus on the three independent claims as amended: claims 1, 11, and 14. Appellants conceded before the Board that dependent claims 2-4 stand or fall with claim 1. Appellants' Br. to Board at 3. Moreover, claims 5- 10 are ultimately dependent on claim 1; claims 12 and 13 are dependent on claim 11; and claims 15-19 are ultimately dependent on claim 14. Although appellants urge that the examiner rejected claims 5-10, 12, 13, and 15-19 without providing any evidence of the additional features of these claims, appellants fail to identify the specific errors in the Board's decision or even the features of these claims that would allow them to overcome an obviousness rejection. Such a conclusory argument by appellants is not sufficient to raise separate issues on appeal with respect to the dependent claims. Thus, all of the dependent claims must stand or fall with their respective independent claims 1, 11, and 14.

The independent claims provide as follows:

1. A voice activated Hypermedia system using grammatical metadata, said system comprising:
a. a speech user agent;
b. a network browsing module; and
c. an information resource located on a computer network wherein said speech user agent facilitates voice activation of said network browsing module to access said information resource.

11. A voice activated Hypermedia system using grammatical metadata, said system comprising:
a. a speech user agent;
b. a network browsing module;
c. an information resource located on a computer network wherein said speech user agent facilitates voice activation

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

of said network browsing module to access said information resource;
d. a means for extracting a grammar from a hypermedia source on said information resource for future reference to said source;
e. a means for modifying said grammar;
f. a means for automatically producing an intelligent grammar from said information resource; and
g. a means for processing said grammar to produce a reference to said hypermedia source.
14. A voice activated Hypermedia system using grammatical metadata, said system comprising:
a. a speech user agent;
b. a browsing module;
c. an information resource; and
d. a means for producing a grammar from textual representation of links to said information resource.
'229 application at 13-14; Jan. 14, 1997 Amendment at 1-3.

The invention recited in claim 1 comprises a speech interface (*i.e.,* the speech user agent) that allows users to access information located on a computer network using a network browsing module, a software tool used to locate database sites on a network. Using the speech interface, an individual can use voice commands to activate the browser and access information located on the computer network, such as a desired Hypertext Markup Language ("HTML") page. The user can access various links on a given HTML page by speaking aloud the link names.

Claims 11 and 14 add the ability to create a "grammar." A grammar is an established set of standard query words. '229 application at 11. According to the '229 application, each user can construct a grammar and associate it with a Uniform ***1361** Resource Location ("URL"), creating a "Speakable Hotlist." *Id.* at 4. A "Speakable Hotlist" allows the user to access a number of different URLs by just saying a phrase. For example, saying the phrase "how does the weather look today" could immediately link the user to the URL < http:// www.washingtonpost.com/weather>. *Id.* at 5.

Claim 11 differs from claim 1 by adding a number of means limitations providing the functions of extracting, modifying, dynamically producing, and processing the grammar. Claim 14 differs from claim 1 by reciting "a means for producing a grammar from textual representation of links to [the] information resource." '229 application at 14.

On August 7, 1996, the examiner issued a first Office Action, which rejected claims 1-19 under "35 U.S.C. § 103 as being unpatentable over Stefanopoulos et al. [U.S. Patent No. 5,333,237] and in view of Schmandt [Chris Schmandt et al., *Augmenting a Window System with Speech Input,* Computer Magazine, Aug. 1990, at 50 ("Schmandt")]...." Aug. 7, 1996 Office Action at 3.

These prior art references relate most directly to claim 1. U.S. Patent No. 5,333,237 (the "'237 patent") issued to Stefanopoulos, et al. ("Stefanopoulos") is entitled "Hypermedia Structured Knowledge Base System." Stefanopoulos is directed to a hypermedia-structured expert system (*i.e.,* a browser software system) that searches and accesses archived electronic documents. '237 patent, col. 1, ll. 11-15. The disclosed invention "incorporates the ability to have both an electronic-based document and an expert system [*i.e.,* a browser software system] coexist within a single computer program." *Id.* at col. 2, ll. 52-54. The browser software system and the electronic documents are interlinked so that the user can move easily between them. *Id.* at col. 2, ll. 55-57. Stefanopoulos discloses the processing of user action based on embedded intelligence and returning results to the user. *Id.* at col. 6, ll. 52-54. Stefanopoulos also discloses an advice (or help) scheme that provides troubleshooting advice to the user, *id.* at col. 6, ll. 55-68, in addition to disclosing the textual representation of hypertext links. *Id.* at col. 6, ll. 32-42, col. 7, ll. 18-38.

Another pertinent prior art reference is an article entitled "Augmenting a Window System with Speech Input," authored by Chris Schmandt, Mark S. Ackerman, and Debby Hindus. The Schmandt reference is directed to controlling a windows system by speech input. Schmandt at 50. Speaking a window's name will move the window to the front of the screen and move the cursor into that window. *Id.* Schmandt describes how speech can be used to navigate in a windows system so that speech actually "assume[s] some of the functions currently assigned to the mouse." *Id.* Schmandt describes two speech interfaces: "Xspeak" and "Xspeak II." *Id.* at 52, 54. Xspeak rearranges windows by "associat[ing] windows with voice *templates,* [*i.e.,*] words trained and stored in the recognizer and constituting its vocabulary." *Id.* at 52. Xspeak II improves on the speech interface of the original Xspeak system by making two changes: (1) "improv[ing] recognition rates ... [by] add[ing] the ability to create subtemplates;" and (2) "includ[ing] a specialized language, G-XL, to facilitate general-purpose handling of the window[s] system." *Id.* at 54. In other words, "Xspeak II allows greater flexibility in the speech interface." *Id.*

On August 7, 1996, the examiner concluded that claim 1 was obvious because Stefanopoulos taught a "hypermedia structured knowledge base system" comprising a browsing module and an information resource, and Schmandt taught a speech interface that served as a navigational tool. ***1362** Aug. 7, 1996 Office Action at 3. The examiner's motivation to combine the two references was that the speech interface to the Web "allows easy access to information on the Web by reducing manual intervention [i.e., the use of keyboard], and ... is user friendly." *Id.*

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

Although the examiner did not find that Stefanopoulos and Schmandt disclosed the grammar-creation capability added in claims 11 and 14, the examiner also rejected both of these claims, noting that "[t]he use of grammar is old and well known in the art of speech recognition as a means of optimization which is highly desirable." *Id.* at 5. The examiner concluded that the additional features specified in the dependent claims were also unpatentable over the cited references, and ultimately rejected claims 1-19 under 35 U.S.C. § 103.

On January 14, 1997, appellants amended claims 1-5, 7-9, and 11 to attempt to overcome the obviousness rejections. Most pertinently, the "information resource" of claim 1 was further defined as "located on a computer network wherein said speech user agent facilitates voice activation of said network browsing module to access said information resource." Jan. 14, 1997 Amendment at 2. This limitation was also added to claim 11 in addition to amending element (d) of claim 11 to recite "a means for extracting a grammar from a hypermedia source *on said information resource* for future reference to said source." *Id.* at 3 (emphasis indicating added language).

On March 27, 1997, in response to the amendment, the examiner issued a final Office Action, maintaining the rejection of claims 1-19 of the '229 application. Mar. 27, 1997 Final Office Action at 2. The examiner found the claims obvious, even in light of the "newly added limitation of '... an information resource located on a computer network ...,' " because Schmandt "teach[es] a window[s] system that links together workstations," and "[i]t would have been obvious to one with ordinary skill in the art at the time of invention that workstations linked together form a network and are served by an information resource, because an artisan would recognize that this is an efficient usage of space available." *Id.* at 2-3.

The Board affirmed the examiner's rejection of all 19 claims. *Decision on Appeal* at 10. The Board concluded with respect to independent claim 1, "the skilled artisan would have been motivated and found it obvious to incorporate the speech input and speech recognition techniques taught by Schmandt into the expert system of Stefanopoulos in order to reduce the need for less user friendly manual keyboard and mouse click inputs." *Id.* at 5. The Board sustained the examiner's rejection of dependent claims 2-4 as falling with claim 1. *Id.* at 8. The Board further noted that "independent claims 11 and 14 are similar to independent claim 1 ... but add limitations directed to the construction and extraction of grammar related features as part of the speech user agent." *Id.* at 9. The Board concluded with respect to claims 5-19, "the Examiner's reasoning is sufficient to shift the burden to Appellants to come forward with evidence and/or arguments to rebut the Examiner's position." *Id.* Although the Board acknowledged that "the applied references do not explicitly disclose several of the various features of claims 5-19," it sustained the rejection of these claims because "Appellants' minimal arguments do not address the Examiner's position of obviousness or inherency with respect to the features recited in these claims." *Id.* at 10.

On rehearing, the Board denied appellants' request to reconsider its previous decision rejecting claims 1-19. *Decision on Request for Rehearing* at 7. The Board once again noted the sufficiency of the ***1363** examiner's reasoning in proving a *prima facie* case of obviousness and the lack of evidence to rebut the examiner's position. *Id.* at 6-7.

We have jurisdiction over this appeal pursuant to 28 U.S.C. § 1295(a)(4)(A).

DISCUSSION

I

[2] "The ultimate determination of whether an invention would have been obvious under 35 U.S.C. § 103(a) is a legal conclusion based on underlying findings of fact." *In re Kotzab,* 217 F.3d 1365, 1369, 55 USPQ2d 1313, 1316 (Fed.Cir.2000). The Board's ultimate determination of obviousness is reviewed without deference; the Board's underlying factual findings are reviewed for substantial evidence. *Id.*

II

Appellants urge that their invention described in the '229 application was not obvious under 35 U.S.C. § 103 over Stefanopoulos in view of Schmandt because (1) the record does not support the Board's determination that the examiner made a *prima facie* case of obviousness of claim 1; and (2) the Board failed to provide evidence of the presence of key limitations of claims 11 and 14 in the prior art. We address each argument in turn.

A. *Prima Facie Case of Obviousness and Claim 1*

[3] To establish a *prima facie* case of obviousness, the Board must, *inter alia,* show "some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references." *In re Fine,* 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed.Cir.1988). "The motivation, suggestion or teaching may come explicitly from statements in the prior art, the knowledge of one of ordinary skill in the art, or, in some cases the nature of the problem to be solved." *Kotzab,* 217 F.3d at 1370, 55 USPQ2d at 1317.

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

[4] The Board agreed that the examiner had satisfied his burden of presenting a *prima facie* case of obviousness with respect to claim 1. *Decision on Appeal* at 5. The Board explained:

> With respect to independent claim 1, the Examiner, as the basis for the obviousness rejection, proposes to modify the disclosed Hypermedia structured expert system of Stefanopoulos which, in the Examiner's view, discloses the claimed invention except for the speech user agent. To address this deficiency, the Examiner turns to Schmandt which discloses the use of speech input to navigate through a multiple window based computer system display. In the Examiner's analysis ... the skilled artisan would have been *motivated and found it obvious to incorporate the speech input and speech recognition techniques taught by Schmandt into the expert system [i.e., here the browser software system] of Stefanopoulos in order to reduce the need for less user friendly manual keyboard and mouse click inputs.*
>
> ... In our view the Examiner's analysis is sufficiently reasonable that we find that the Examiner has at least satisfied the burden of presenting a *prima facie* case of obviousness.

Decision on Appeal at 4-5 (emphasis added).

Appellants appear to concede that Stefanopoulos discloses all of the limitations of the claimed invention except the "speech user agent." However, appellants urge that Schmandt does not disclose a "speech user agent." Appellants define a "speech user agent" as an agent that "*dynamically creates the vocabulary, grammar and actions* that are possible for the user to use ***1364** in a given situation." Appellants' Br. at 35 (emphasis added).

[5] When examining claims for patentability, claims are interpreted as broadly as is reasonable and consistent with the specification. *See, e.g., In re Hyatt,* 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed.Cir.2000). Because claim 1 uses the general term "speech user agent" without any modifiers, appellants' limiting interpretation of this phrase is inappropriate, particularly in light of the grammar-creation capability separately recited in claims 11 and 14. Instead, we interpret the phrase "speech user agent" consistent with its plain meaning: an interface that allows the user to interact with the system by speaking. *See IBM Dictionary of Computing* 638 (10th ed.1994) (defining "speech recognition" as "[t]he recognition of voice communication as a series of words or sentences"); *Van Nostrand Reinhold Dictionary of Information Technology* 473 (3d ed.1989) (defining "speech recognizer" as "[i]n man-machine interfaces, a system that receives spoken word inputs and identifies the message. The system output can then be used to initiate appropriate actions or responses."); *Johnson Worldwide Assocs., Inc. v. Zebco Corp.,* 175 F.3d 985, 989, 50 USPQ2d 1607, 1610 (Fed.Cir.1999) ("The general rule is, of course, that terms in the claim are to be given their ordinary and accustomed meaning."). Although an applicant may be his own lexicographer, *Markman v. Westview Instruments, Inc.,* 52 F.3d 967, 980, 34 USPQ2d 1321, 1330 (Fed.Cir.1995) (*en banc*), *aff'd,* 517 U.S. 370, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996), nothing in the specification defines the phrase "speech user agent" differently from its ordinary meaning. Applying the above definition, Schmandt clearly discloses a "speech user agent" because the reference specifically describes "Xspeak," a speech interface with the windows environment.

[6] Appellants also argue that even if Schmandt discloses a "speech user agent," there is no suggestion or motivation to combine Schmandt and Stefanopoulos. Recently, in *In re Lee,* 277 F.3d 1338, 61 USPQ2d 1430 (Fed.Cir.2002), we held that the Board's reliance on "common knowledge and common sense" did not fulfill the agency's obligation to cite references to support its conclusions. *Id.* at 1344, 277 F.3d 1338, 61 USPQ2d at 1434. Instead, the Board must document its reasoning on the record to allow accountability. *Id.* at 1345, 277 F.3d 1338, 61 USPQ2d at 1435. This documentation also allows effective judicial review. *Id.*

[7] In the present case, the reasoning articulated by the Board is exactly the type of reasoning required by *In re Lee.* Both the examiner and the Board clearly identified a motivation to combine the references, stating that the skilled artisan would have "found it obvious to incorporate the speech input and speech recognition techniques taught by Schmandt into the expert system of Stefanopoulos in order to reduce the need for less user friendly manual keyboard and mouse click inputs." *Decision on Appeal* at 5; *accord* Aug. 7, 1996 Office Action at 3. The motivation to combine the references is present in the text of each reference. The Schmandt reference itself verifies this motivation, stating that "allowing users to remain focused on the screen and keyboard, instead of fumbling for the mouse, would be beneficial in a workstation environment." *Schmandt* at 51. Stefanopoulos itself, while not expressly disclosing the use of speech recognition, sets forth the motivation to combine the references, stating that "there are alternative means to select the buttons, including ... voice-activated transfer means, which may be readily adapted for use with the present invention ***1365** by those skilled in the art." '237 patent, col. 4, ll. 34-38.

Appellants also argue that Schmandt discloses only a pre-programmed grammar. Because the construction of "speech user agent" does not require an agent that dynamically changes grammar and vocabulary, but only an agent that allows the user to interact with the system by speaking, we conclude that Schmandt adequately discloses a speech user agent as required by claim 1 of the '229

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

application. Appellants further urge that Schmandt's invention is directed to navigation in a windows system, not the web-browsing environment of the present invention. However, the Board properly concluded that the present invention is not limited to a web-browsing environment, stating that "the language of the appealed claims reveals only a recitation of a 'computer network' with no recitation of any remote accessing of a database or any limitation as to database complexity or the internet." *Decision on Appeal* at 8. Appellants also argue that the prior art references revealed no reasonable expectation of success in carrying out the invention; however, the Board found that this issue had not been properly raised. Finally, we are not persuaded that Schmandt does not suggest the combination because it only refers to speech recognition as substituting for the mouse and not the keyboard. Accordingly, the Board's conclusion that the invention recited in claim 1 was obvious is well supported.

For these reasons, we affirm the Board's decision as to independent claim 1 and dependent claims 2-10.

B. *Evidence of Key Limitations of Claims 11 and 14 in the Prior Art*

1. Claim 11

[8] As noted above, claim 11 differs from claim 1 in that claim 11 adds grammar-creation capability features including a means for extracting a grammar from a hypermedia source; a means for modifying the grammar; a means for dynamically producing the grammar; and a means for processing the grammar to produce a reference to a specific hypermedia source. In the August 1996 Office Action, the examiner summarily rejected claim 11's grammar-creation capability limitations, stating that "[t]he use of grammar is old and well known in the art of speech recognition as a means of optimization which is highly desirable." Aug. 7, 1996 Office Action at 5. The examiner's final Office Action generally affirmed his initial conclusion as to obviousness, without specifically discussing the grammar features. Mar. 27, 1997 Final Office Action at 2. The Board subsequently affirmed the examiner's decision to reject claim 11, noting that the examiner set forth sufficient reasoning for asserting the obviousness or inherent nature of each of the claimed features, and therefore, "the Examiner's reasoning is sufficient to shift the burden to Appellants to come forward with evidence and/or arguments to rebut the Examiner's position." *Decision on Appeal* at 9; *accord Decision on Request for Rehearing* at 6.

Appellants urge that this rejection was improper because there is no evidence in the record to prove that the following limitations of claim 11 would be obvious:

> d. a means for extracting a grammar from a hypermedia source on said information resource for future reference to said source;
> e. a means for modifying said grammar;
>
> g. a means for processing said grammar to produce a reference to said hypermedia source.

Even if all of the limitations of claim 11 were present in the prior art, appellants urge that there is no teaching or suggestion ***1366** supporting the combination of these references.

We agree with appellants that the Board's ground of rejection is simply inadequate on its face. The Board sustained the examiner's very general and broad conclusion of obviousness based on his finding that "[t]he use of grammar is old and well known in the art of speech recognition as a means of optimization which is highly desirable." Aug. 7, 1996 Office Action at 5; *accord Decision on Request for Rehearing* at 6. Although this statement is likely true, it fails to address the grammar-creation capability limitations of claim 11. While the examiner's statement generally addresses the use of grammar, it does not discuss the unique limitations of extracting, modifying, or processing the grammar to interact with hypermedia sources. The Board's decision is not supported by substantial evidence because the cited references do not support each limitation of claim 11. *See In re Vaeck*, 947 F.2d 488, 493, 20 USPQ2d 1438, 1443 (Fed.Cir.1991).

[9] The Patent Office ("PTO") on appeal does not even defend the Board's rejection of claim 11 on the merits. Rather, it urges that appellants did not properly raise the grounds for sustaining the patent before the Board. The PTO's argument seems weak at best because appellants' brief to the Board specifically discusses the alleged failure of the Schmandt and Stefanopoulos references to teach the claimed invention of claim 11. Despite language in the Board's reconsideration decision faulting appellants for failure to raise issues before the Board, [FN1] we do not read the Board's decisions on appeal or reconsideration as resting on this ground. Rather, the Board sustained the examiner's rejection on the merits.

> FN1. The Board noted that "neither of [Appellants' arguments regarding motivation were] made by Appellants in the Brief. An argument not timely made is an argument waived." *Decision on Request for Rehearing* at 4.

In *SEC v. Chenery Corp.*, 332 U.S. 194, 196, 67 S.Ct. 1575, 91 L.Ed. 1995 (1947), the Supreme Court held that

> a reviewing court, in dealing with a determination or judgment which an administrative agency alone is authorized to make, must judge the propriety of such action solely by the grounds invoked by the agency. If

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

those grounds are inadequate or improper, the court is powerless to affirm the administrative action by substituting what it considers to be a more adequate or proper basis.

Here the *Chenery* rule is implicated because the Board failed to provide an adequate ground for sustaining the rejection of claim 11 in either its decision on appeal or its decision on request for rehearing. *See also SKF USA Inc. v. United States,* 254 F.3d 1022, 1028 (Fed.Cir.2001).

Therefore, because the Board did not hold that appellants' arguments were untimely and did reach the merits of claim 11 but failed to provide an adequate basis for rejecting this claim, we vacate and remand for further proceedings before the Board as to independent claim 11 and dependent claims 12 and 13.

2. Claim 14

Since the limitations of claims 11 and 14 are very similar, the examiner rejected claim 14 on the same grounds as he rejected claim 11. Appellants urge that there was also insufficient evidence in the record to support the examiner's and Board's conclusions regarding obviousness because the cited references do not support all of the limitations of claim 14. We agree that the Board's decision is not supported by substantial evidence. Again, in its brief to this court, the PTO attempts to sustain the actions of the Board on grounds not articulated ***1367** in the Board's decision, relying on Schmandt to defend the rejection of claim 14. The PTO claims that Schmandt's Xspeak II system discloses the grammar limitation of claim 14: "a means for producing a grammar from textual representation of links to said information resource." The PTO further contends that Schmandt provides the motivation to combine a speech user agent such as Xspeak II with the information resource described in Stefanopoulos because Xspeak II provides the benefits of decreased manual input and improved recognition rates.

Whatever merit there is to the PTO's defense of the rejection of claim 14 raised in its brief submitted to this court, that ground does not appear in the Board's decision and may not be the basis for affirmance. *See Chenery,* 332 U.S. at 196, 67 S.Ct. 1575; *In re De Blauwe,* 736 F.2d 699, 705 n. 7, 222 USPQ 191, 196 n. 7 (Fed.Cir.1984) ("[T]he Solicitor cannot raise a new ground of rejection or apply a new rationale to support a rejection in appeals from decisions of the board."). We vacate and remand for further proceedings as to independent claims 14 and dependent claims 15-19.

CONCLUSION

For the foregoing reasons, we affirm the Board's decision as to claims 1-10, and we vacate and remand the Board's decision as to claims 11-19 for further proceedings consistent with this opinion.

COSTS

No costs.

298 F.3d 1357, 63 U.S.P.Q.2d 2002

END OF DOCUMENT

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

United States Court of Appeals,
Federal Circuit.

In re SANG-SU LEE.

No. 00-1158.

Jan. 18, 2002.

Board of Patent Appeals and Interferences rejected all claims of inventor's patent application directed toward method of automatically displaying functions of video display device that demonstrated how to select and adjust functions in order to facilitate response by user. Inventor appealed. The Court of Appeals, Pauline Newman, Circuit Judge, held that analysis by Board did not comport with either legal requirements for determination of obviousness or with requirements of Administrative Procedure Act (APA).

Vacated and remanded.

West Headnotes

[1] Patents ⚖113(6)
291k113(6) Most Cited Cases
Tribunals of the Patent and Trademark Office (PTO) are governed by the Administrative Procedure Act (APA), and their rulings receive the same judicial deference as do tribunals of other administrative agencies. 5 U.S.C.A. § 551 et seq.

[2] Administrative Law and Procedure ⚖485
15Ak485 Most Cited Cases

[2] Administrative Law and Procedure ⚖507
15Ak507 Most Cited Cases

For judicial review to be meaningfully achieved within the strictures of the Administrative Procedures Act (APA), an agency tribunal must present a full and reasoned explanation of its decision; the agency tribunal must set forth its findings and the grounds thereof, as supported by the agency record, and explain its application of the law to the found facts. 5 U.S.C.A. § 706(2).

[3] Patents ⚖113(6)
291k113(6) Most Cited Cases

Judicial review of a decision of the Board of Patent Appeals and Interferences denying an application for a patent is founded on the obligation of the agency to make the necessary findings and to provide an administrative record showing the evidence on which the findings are based, accompanied by the agency's reasoning in reaching its conclusions. 5 U.S.C.A. § 551 et seq.

[4] Patents ⚖31.1
291k31.1 Most Cited Cases

As applied to the determination of patentability vel non when the issue is obviousness, it is fundamental that the rejection of a patent application must be based on evidence comprehended by the language of the statute addressing obviousness. 35 U.S.C.A. § 103.

[5] Patents ⚖16.5(1)
291k16.5(1) Most Cited Cases

The patent examination process centers on prior art and the analysis thereof; when patentability turns on the question of obviousness, the search for and analysis of the prior art includes evidence relevant to the finding of whether there is a teaching, motivation, or suggestion to select and combine the references relied on as evidence of obviousness. 35 U.S.C.A. § 103.

[6] Patents ⚖26(1)
291k26(1) Most Cited Cases

In the context of an obviousness determination by the Board of Patent Appeals and Interferences, the factual inquiry whether to combine references must be thorough and searching; it must be based on objective evidence of record. 35 U.S.C.A. § 103.

[7] Patents ⚖111
291k111 Most Cited Cases

Analysis of invention by Board of Patent Appeals and Interferences did not comport with either legal requirements for determination of obviousness or with requirements of Administrative Procedure Act (APA) on basis that agency tribunal did not set forth findings and explanations needed for reasoned decisionmaking; examiner used conclusory statements to support his subjective belief that it was obvious that person skilled in the art would have been motivated to combine prior art, and Board rejected need for any specific hint or suggestion in particular reference to support combination of prior art. 5 U.S.C.A. § 706(2); 35 U.S.C.A. § 103.

[8] Patents ⚖26(1)
291k26(1) Most Cited Cases

In an obviousness determination, the factual question of motivation to combine prior art is material to patentability, and cannot be resolved on subjective belief and unknown authority. 35 U.S.C.A. § 103.

[9] Patents ⚖26(1)
291k26(1) Most Cited Cases

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

[9] Patents ⚖111
291k111 Most Cited Cases

In an obviousness determination under patent law, it is improper, in determining whether a person of ordinary skill would have been led to combine references, simply to use that which the inventor taught against its teacher; thus, the Board of Patent Appeals and Interferences must not only assure that the requisite findings are made, based on evidence of record, but must also explain the reasoning by which the findings are deemed to support the agency's conclusion. 5 U.S.C.A. § 706(2); 35 U.S.C.A. § 103.

[10] Administrative Law and Procedure ⚖507
15Ak507 Most Cited Cases

Deferential judicial review under the Administrative Procedure Act (APA) does not relieve the agency of its obligation to develop an evidentiary basis for its findings; to the contrary, the APA reinforces this obligation. 5 U.S.C.A. § 706(2).

[11] Administrative Law and Procedure ⚖763
15Ak763 Most Cited Cases

[11] Administrative Law and Procedure ⚖796
15Ak796 Most Cited Cases

In the context of judicial review under the Administrative Procedure Act (APA), a decision by an agency tribunal that has an omission of a relevant factor required by precedent is both legal error and "arbitrary agency action." 5 U.S.C.A. § 551 et seq.

[12] Administrative Law and Procedure ⚖485
15Ak485 Most Cited Cases

[12] Administrative Law and Procedure ⚖760
15Ak760 Most Cited Cases

The foundation of the principle of judicial deference under the Administrative Procedures Act (APA) to the rulings of agency tribunals is that the tribunal has specialized knowledge and expertise, such that when reasoned findings are made, a reviewing court may confidently defer to the agency's application of its knowledge in its area of expertise; however, reasoned findings are critical to the performance of agency functions and judicial reliance on agency competence. 5 U.S.C.A. § 706(2).

[13] Patents ⚖16(1)
291k16(1) Most Cited Cases

The determination of patentability on the ground of unobviousness is ultimately one of judgment; in furtherance of the judgmental process, the patent examination procedure serves both to find, and to place on the official record, that which has been considered with respect to patentability. 35 U.S.C.A. § 103.

[14] Patents ⚖16(3)
291k16(3) Most Cited Cases

[14] Patents ⚖104
291k104 Most Cited Cases

[14] Patents ⚖111
291k111 Most Cited Cases

In the context of an obviousness determination, the patent examiner and the Board of Patent Appeals and Interferences are deemed to have experience in the field of the invention; however, this experience, insofar as applied to the determination of patentability, must be applied from the viewpoint of the person having ordinary skill in the art to which said subject matter pertains. 35 U.S.C.A. § 103.

[15] Patents ⚖104
291k104 Most Cited Cases

In finding the relevant facts, in assessing the significance of the prior art, and in making the ultimate determination of the issue of obviousness, the examiner and the Board of Patent Appeals and Interferences are presumed to act from the viewpoint of a person having ordinary skill in the art to which the subject matter pertains; thus, when they rely on what they assert to be general knowledge to negate patentability, that knowledge must be articulated and placed on the record and the failure to do so is not consistent with either effective administrative procedure or effective judicial review. 5 U.S.C.A. § 706(2); 35 U.S.C.A. § 103.

[16] Patents ⚖111
291k111 Most Cited Cases

In the context of an obviousness determination, the Board of Patent Appeals and Interferences cannot rely on conclusory statements when dealing with particular combinations of prior art and specific claims, but must set forth the rationale on which it relies. 5 U.S.C.A. § 706(2); 35 U.S.C.A. § 103.

[17] Administrative Law and Procedure ⚖326
15Ak326 Most Cited Cases

[17] Administrative Law and Procedure ⚖485
15Ak485 Most Cited Cases

[17] Administrative Law and Procedure ⚖507
15Ak507 Most Cited Cases

Sound administrative procedure requires that an agency apply the law in accordance with statute and precedent; the

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

agency tribunal must make findings of relevant facts, and present its reasoning in sufficient detail that the court may conduct meaningful review of the agency action. 5 U.S.C.A. § 706(2).

Patents ⇐328(2)
291k328(2) Most Cited Cases

4,626,892. Cited As Prior Art.
***1340** Richard H. Stern, of Washington, DC, argued for Sang Su Lee. With him on the brief was Robert E. Bushnell.

Sidney O. Johnson, Jr., Associate Solicitor, of Arlington, Virginia, argued for the Director of the U.S. Patent and Trademark Office. With him on the brief were John M. Whealan, Solicitor, and Raymond T. Chen, Associate Solicitor. Of counsel were Maximilian R. Peterson and Mark Nagumo, Associate Solicitors.

Before PAULINE NEWMAN, CLEVENGER, and DYK, Circuit Judges.

PAULINE NEWMAN, Circuit Judge.

Sang-Su Lee appeals the decision of the Board of Patent Appeals and Interferences of the United States Patent and Trademark Office, rejecting all of the claims of Lee's patent application Serial No. 07/631,210 entitled "Self- Diagnosis and Sequential-Display Method of Every Function." [FN1] We vacate the Board's decision for failure to meet the adjudicative standards for review under the Administrative Procedure Act, and remand for further proceedings.

> FN1. *Ex parte Lee*, No.1994-1989 (Bd. Pat.App. & Int. Aug. 30, 1994; on reconsid'n Sept. 29, 1999).

The Prosecution Record

Mr. Lee's patent application is directed to a method of automatically displaying the functions of a video display device and demonstrating how to select and adjust the functions in order to facilitate response by the user. The display and demonstration are achieved using computer-managed electronics, including pulse-width modulation and auto-fine-tuning pulses, in accordance with procedures described in the specification. Claim 10 is representative:

> 10. A method for automatically displaying functions of a video display device, comprising:
> determining if a demonstration mode is selected;
> if said demonstration mode is selected, automatically entering a picture adjustment mode having a picture menu screen displaying a list of a plurality of picture functions; and
> automatically demonstrating selection and adjustment of individual ones of said plurality of picture functions.

The examiner rejected the claims on the ground of obviousness, citing the combination of two references: United States Patent No. 4,626,892 to Nortrup, and the Thunderchopper Helicopter Operations ***1341** Handbook for a video game. The Nortrup reference describes a television set having a menu display by which the user can adjust various picture and audio functions; however, the Nortrup display does not include a demonstration of how to adjust the functions. The Thunderchopper Handbook describes the Thunderchopper game's video display as having a "demonstration mode" showing how to play the game; however, the Thunderchopper Handbook makes no mention of the adjustment of picture or audio functions. The examiner held that it would have been obvious to a person of ordinary skill to combine the teachings of these references to produce the Lee system.

Lee appealed to the Board, arguing that the Thunderchopper Handbook simply explained how to play the Thunderchopper game, and that the prior art provided no teaching or motivation or suggestion to combine this reference with Nortrup, or that such combination would produce the Lee invention. The Board held that it was not necessary to present a source of a teaching, suggestion, or motivation to combine these references or their teachings. The Board stated:

> The conclusion of obviousness may be made from common knowledge and common sense of a person of ordinary skill in the art without any specific hint or suggestion in a particular reference.

Board op. at 7. The Board did not explain the "common knowledge and common sense" on which it relied for its conclusion that "the combined teachings of Nortrup and Thunderchopper would have suggested the claimed invention to those of ordinary skill in the art."

Lee filed a request for reconsideration, to which the Board responded after five years. The Board reaffirmed its decision, stating that the Thunderchopper Handbook was "analogous art" because it was "from the same field of endeavor" as the Lee invention, and that the field of video games was "reasonably pertinent" to the problem of adjusting display functions because the Thunderchopper Handbook showed video demonstrations of the "features" of the game. On the matter of motivation to combine the Nortrup and Thunderchopper references, the Board stated that "we maintain the position that we stated in our prior decision" and that the Examiner's Answer provided "a well reasoned discussion of why there is sufficient motivation to combine the references." The Board did not state the examiner's reasoning, and review of the Examiner's Answer reveals that the examiner merely stated that both the Nortrup function menu and the Thunderchopper demonstration mode are program features and that the

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

Thunderchopper mode "is user-friendly" and it functions as a tutorial, and that it would have been obvious to combine them.

Lee had pressed the examiner during prosecution for some teaching, suggestion, or motivation in the prior art to select and combine the references that were relied on to show obviousness. The Examiner's Answer before the Board, plus a Supplemental Answer, stated that the combination of Thunderchopper with Nortrup "would have been obvious to one of ordinary skill in the art since the demonstration mode is just a programmable feature which can be used in many different device[s] for providing automatic introduction by adding the proper programming software," and that "another motivation would be that the automatic demonstration mode is user friendly and it functions as a tutorial." The Board adopted the examiner's answer, stating "the examiner has provided a well reasoned discussion of these references and how the combination of these references meets the claim limitations." However, perhaps recognizing that the examiner had provided insufficient justification to ***1342** support combining the Nortrup and Thunderchopper references, the Board held, as stated *supra,* that a "specific hint or suggestion" of motivation to combine was not required.

This appeal followed.

Judicial Review

[1] Tribunals of the PTO are governed by the Administrative Procedure Act, and their rulings receive the same judicial deference as do tribunals of other administrative agencies. *Dickinson v. Zurko,* 527 U.S. 150, 119 S.Ct. 1816, 144 L.Ed.2d 143, 50 USPQ2d 1930 (1999). Thus on appeal we review a PTO Board's findings and conclusions in accordance with the following criteria:

> 5 U.S.C. § 706(2) The reviewing court shall--
> (2) hold unlawful and set aside agency actions, findings, and conclusions found to be--
> (A) arbitrary, capricious, an abuse of discretion, or otherwise not in accordance with law;
>
> * * * * * *
>
> (E) unsupported by substantial evidence in a case subject to sections 556 and 557 of this title or otherwise reviewed on the record of an agency hearing provided by statute;

[2][3] For judicial review to be meaningfully achieved within these strictures, the agency tribunal must present a full and reasoned explanation of its decision. The agency tribunal must set forth its findings and the grounds thereof, as supported by the agency record, and explain its application of the law to the found facts. The Court has often explained:

> The Administrative Procedure Act, which governs the proceedings of administrative agencies and related judicial review, establishes a scheme of "reasoned decisionmaking." Not only must an agency's decreed result be within the scope of its lawful authority, but the process by which it reaches that result must be logical and rational.

Allentown Mack Sales and Service, Inc. v. National Labor Relations Bd., 522 U.S. 359, 374, 118 S.Ct. 818, 139 L.Ed.2d 797 (1998) (citation omitted). This standard requires that the agency not only have reached a sound decision, but have articulated the reasons for that decision. The reviewing court is thus enabled to perform meaningful review within the strictures of the APA, for the court will have a basis on which to determine "whether the decision was based on the relevant factors and whether there has been a clear error of judgment." *Citizens to Preserve Overton Park v. Volpe,* 401 U.S. 402, 416, 91 S.Ct. 814, 28 L.Ed.2d 136 (1971). Judicial review of a Board decision denying an application for patent is thus founded on the obligation of the agency to make the necessary findings and to provide an administrative record showing the evidence on which the findings are based, accompanied by the agency's reasoning in reaching its conclusions. *See In re Zurko,* 258 F.3d 1379, 1386, 59 USPQ2d 1693, 1697 (Fed.Cir.2001) (review is on the administrative record); *In re Gartside,* 203 F.3d 1305, 1314, 53 USPQ2d 1769, 1774 (Fed.Cir.2000) (Board decision "must be justified within the four corners of the record").

[4][5] As applied to the determination of patentability *vel non* when the issue is obviousness, "it is fundamental that rejections under 35 U.S.C. § 103 must be based on evidence comprehended by the language of that section." *In re Grasselli,* 713 F.2d 731, 739, 218 USPQ 769, 775 (Fed.Cir.1983). The essential factual evidence on the issue of obviousness is set forth in *Graham v. John Deere Co.,* 383 U.S. 1, 17-18, 86 S.Ct. 684, 15 L.Ed.2d 545, 148 USPQ 459, 467 (1966) and extensive ensuing precedent. The patent examination ***1343** process centers on prior art and the analysis thereof. When patentability turns on the question of obviousness, the search for and analysis of the prior art includes evidence relevant to the finding of whether there is a teaching, motivation, or suggestion to select and combine the references relied on as evidence of obviousness. *See, e.g., McGinley v. Franklin Sports, Inc.,* 262 F.3d 1339, 1351-52, 60 USPQ2d 1001, 1008 (Fed.Cir.2001) ("the central question is whether there is reason to combine [the] references," a question of fact drawing on the *Graham* factors).

[6] "The factual inquiry whether to combine references must be thorough and searching." *Id.* It must be based on objective evidence of record. This precedent has been reinforced in myriad decisions, and cannot be dispensed with. *See, e.g., Brown & Williamson Tobacco Corp. v.*

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

Philip Morris Inc., 229 F.3d 1120, 1124-25, 56 USPQ2d 1456, 1459 (Fed.Cir.2000) ("a showing of a suggestion, teaching, or motivation to combine the prior art references is an 'essential component of an obviousness holding' ") (quoting *C.R. Bard, Inc., v. M3 Systems, Inc.*, 157 F.3d 1340, 1352, 48 USPQ2d 1225, 1232 (Fed.Cir.1998)); *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed.Cir.1999) ("Our case law makes clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references."); *In re Dance*, 160 F.3d 1339, 1343, 48 USPQ2d 1635, 1637 (Fed.Cir.1998) (there must be some motivation, suggestion, or teaching of the desirability of making the specific combination that was made by the applicant); *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ2d 1596, 1600 (Fed.Cir.1988) (" 'teachings of references can be combined *only* if there is some suggestion or incentive to do so.' ") (emphasis in original) (quoting *ACS Hosp. Sys., Inc. v. Montefiore Hosp.*, 732 F.2d 1572, 1577, 221 USPQ 929, 933 (Fed.Cir.1984)).

The need for specificity pervades this authority. *See, e.g., In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed.Cir.2000) ("particular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed"); *In re Rouffet*, 149 F.3d 1350, 1359, 47 USPQ2d 1453, 1459 (Fed.Cir.1998) ("even when the level of skill in the art is high, the Board must identify specifically the principle, known to one of ordinary skill, that suggests the claimed combination. In other words, the Board must explain the reasons one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious."); *In re Fritch*, 972 F.2d 1260, 1265, 23 USPQ2d 1780, 1783 (Fed.Cir.1992) (the examiner can satisfy the burden of showing obviousness of the combination "only by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references").

[7][8][9] With respect to Lee's application, neither the examiner nor the Board adequately supported the selection and combination of the Nortrup and Thunderchopper references to render obvious that which Lee described. The examiner's conclusory statements that "the demonstration mode is just a programmable feature which can be used in many different device[s] for providing automatic introduction by adding the proper programming software" and that "another motivation would be that the automatic demonstration mode is user friendly and it functions as a tutorial" do not adequately address the issue of motivation to combine. This factual question ***1344** of motivation is material to patentability, and could not be resolved on subjective belief and unknown authority. It is improper, in determining whether a person of ordinary skill would have been led to this combination of references, simply to "[use] that which the inventor taught against its teacher." *W.L. Gore v. Garlock, Inc.*, 721 F.2d 1540, 1553, 220 USPQ 303, 312-13 (Fed.Cir.1983). Thus the Board must not only assure that the requisite findings are made, based on evidence of record, but must also explain the reasoning by which the findings are deemed to support the agency's conclusion.

[10] Deferential judicial review under the Administrative Procedure Act does not relieve the agency of its obligation to develop an evidentiary basis for its findings. To the contrary, the Administrative Procedure Act reinforces this obligation. *See, e.g., Motor Vehicle Manufacturers Ass'n v. State Farm Mutual Automobile Ins. Co.*, 463 U.S. 29, 43, 103 S.Ct. 2856, 77 L.Ed.2d 443 (1983) ("the agency must examine the relevant data and articulate a satisfactory explanation for its action including a 'rational connection between the facts found and the choice made.' ") (quoting *Burlington Truck Lines v. United States*, 371 U.S. 156, 168, 83 S.Ct. 239, 9 L.Ed.2d 207 (1962)); *Securities & Exchange Comm'n v. Chenery Corp.*, 318 U.S. 80, 94, 63 S.Ct. 454, 87 L.Ed. 626 (1943) ("The orderly function of the process of review requires that the grounds upon which the administrative agency acted are clearly disclosed and adequately sustained.").

[11] In its decision on Lee's patent application, the Board rejected the need for "any specific hint or suggestion in a particular reference" to support the combination of the Nortrup and Thunderchopper references. Omission of a relevant factor required by precedent is both legal error and arbitrary agency action. *See Motor Vehicle Manufacturers*, 463 U.S. at 43, 103 S.Ct. 2856 ("an agency rule would be arbitrary and capricious if the agency ... entirely failed to consider an important aspect of the problem"); *Mullins v. Department of Energy*, 50 F.3d 990, 992 (Fed.Cir.1995) ("It is well established that agencies have a duty to provide reviewing courts with a sufficient explanation for their decisions so that those decisions may be judged against the relevant statutory standards, and that failure to provide such an explanation is grounds for striking down the action."). As discussed in *National Labor Relations Bd. v. Ashkenazy Property Mgt. Corp.*, 817 F.2d 74, 75 (9th Cir.1987), an agency is "not free to refuse to follow circuit precedent."

[12] The foundation of the principle of judicial deference to the rulings of agency tribunals is that the tribunal has specialized knowledge and expertise, such that when reasoned findings are made, a reviewing court may confidently defer to the agency's application of its knowledge in its area of expertise. Reasoned findings are critical to the performance of agency functions and judicial

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

reliance on agency competence. *See Baltimore and Ohio R.R. Co. v. Aberdeen & Rockfish R.R. Co.*, 393 U.S. 87, 91-92, 89 S.Ct. 280, 21 L.Ed.2d 219 (1968) (absent reasoned findings based on substantial evidence effective review would become lost "in the haze of so-called expertise"). The "common knowledge and common sense" on which the Board relied in rejecting Lee's application are not the specialized knowledge and expertise contemplated by the Administrative Procedure Act. Conclusory statements such as those here provided do not fulfill the agency's obligation. This court explained in *Zurko*, 258 F.3d at 1385, 59 USPQ2d at 1697, that "deficiencies of the cited references cannot be remedied by the Board's general conclusions about what is 'basic knowledge' or 'common sense.' " The ***1345** Board's findings must extend to all material facts and must be documented on the record, lest the "haze of so-called expertise" acquire insulation from accountability. "Common knowledge and common sense," even if assumed to derive from the agency's expertise, do not substitute for authority when the law requires authority. *See Allentown Mack*, 522 U.S. at 376, 118 S.Ct. 818 ("Because reasoned decisionmaking demands it, and because the systemic consequences of any other approach are unacceptable, the Board must be required to apply in fact the clearly understood legal standards that it enunciates in principle....")

The case on which the Board relies for its departure from precedent, *In re Bozek*, 57 C.C.P.A. 713, 416 F.2d 1385, 163 USPQ 545 (1969), indeed mentions "common knowledge and common sense," the CCPA stating that the phrase was used by the Solicitor to support the Board's conclusion of obviousness based on evidence in the prior art. *Bozek* did not hold that common knowledge and common sense are a substitute for evidence, but only that they may be applied to analysis of the evidence. *Bozek* did not hold that objective analysis, proper authority, and reasoned findings can be omitted from Board decisions. Nor does *Bozek*, after thirty-two years of isolation, outweigh the dozens of rulings of the Federal Circuit and the Court of Customs and Patent Appeals that determination of patentability must be based on evidence. This court has remarked, in *Smiths Industries Medical Systems, Inc. v. Vital Signs, Inc.*, 183 F.3d 1347, 1356, 51 USPQ2d 1415, 1421 (Fed.Cir.1999), that *Bozek* 's reference to common knowledge "does not in and of itself make it so" absent evidence of such knowledge.

[13][14][15][16] The determination of patentability on the ground of unobviousness is ultimately one of judgment. In furtherance of the judgmental process, the patent examination procedure serves both to find, and to place on the official record, that which has been considered with respect to patentability. The patent examiner and the Board are deemed to have experience in the field of the invention; however, this experience, insofar as applied to the determination of patentability, must be applied from the viewpoint of "the person having ordinary skill in the art to which said subject matter pertains," the words of section 103. In finding the relevant facts, in assessing the significance of the prior art, and in making the ultimate determination of the issue of obviousness, the examiner and the Board are presumed to act from this viewpoint. Thus when they rely on what they assert to be general knowledge to negate patentability, that knowledge must be articulated and placed on the record. The failure to do so is not consistent with either effective administrative procedure or effective judicial review. The board cannot rely on conclusory statements when dealing with particular combinations of prior art and specific claims, but must set forth the rationale on which it relies.

Alternative Grounds

At oral argument the PTO Solicitor proposed alternative grounds on which this court might affirm the Board's decision. However, as stated in *Burlington Truck Lines, Inc. v. United States*, 371 U.S. 156, 168, 83 S.Ct. 239, 9 L.Ed.2d 207 (1962), "courts may not accept appellate counsel's *post hoc* rationalization for agency action." Consideration by the appellate tribunal of new agency justifications deprives the aggrieved party of a fair opportunity to support its position; thus review of an administrative decision must be made on the grounds relied on by the agency. "If those grounds are inadequate or improper, the court is powerless to affirm the administrative action by substituting what it considers ***1346** to be a more adequate or proper basis." *Securities & Exchange Comm'n v. Chenery Corp.*, 332 U.S. 194, 196, 67 S.Ct. 1575, 91 L.Ed. 1995 (1947). As reiterated in *Federal Election Comm'n v. Akins*, 524 U.S. 11, 25, 118 S.Ct. 1777, 141 L.Ed.2d 10 (1998), "If a reviewing court agrees that the agency misinterpreted the law, it will set aside the agency's action and remand the case--even though the agency (like a new jury after a mistrial) might later, in the exercise of its lawful discretion, reach the same result for a different reason." Thus we decline to consider alternative grounds that might support the Board's decision.

Further Proceedings

[17] Sound administrative procedure requires that the agency apply the law in accordance with statute and precedent. The agency tribunal must make findings of relevant facts, and present its reasoning in sufficient detail that the court may conduct meaningful review of the agency action. In *Radio- Television News Directors Ass'n v. FCC*, 184 F.3d 872 (D.C.Cir.1999) the court discussed the "fine line between agency reasoning that is 'so crippled as to be unlawful' and action that is potentially lawful but insufficiently or inappropriately explained," quoting from

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works

Checkosky v. Securities & Exch. Comm'n, 23 F.3d 452, 464 (D.C.Cir.1994); the court explained that "[i]n the former circumstance, the court's practice is to vacate the agency's order, while in the latter the court frequently remands for further explanation (including discussion of the relevant factors and precedents) while withholding judgment on the lawfulness of the agency's proposed action." *Id.* at 888. In this case the Board's analysis of the Lee invention does not comport with either the legal requirements for determination of obviousness or with the requirements of the Administrative Procedure Act that the agency tribunal set forth the findings and explanations needed for "reasoned decisionmaking." Remand for these purposes is required. *See Overton Park,* 401 U.S. at 420- 421, 91 S.Ct. 814 (remanding for further proceedings appropriate to the administrative process).

VACATED AND REMANDED.

277 F.3d 1338, 61 U.S.P.Q.2d 1430

END OF DOCUMENT

Copr. © West 2003 No Claim to Orig. U.S. Govt. Works